ACSM1

efesotomasyon.com

Firmware Manual
ACSM1 Speed and Torque Control Program

ACSM1 Speed and Torque Control Program

Firmware Manual

efesotomasyon.com

3AFE68848261 REV F
EN
EFFECTIVE: 2009-12-21

© 2009 ABB Oy. All Rights Reserved.

Table of contents

Table of contents

Introduction to the manual

What this chapter contains . . . 13
Compatibility . . . 13
Safety instructions . . . 13
Reader . . . 13
Contents . . . 14
Product and service inquiries . . . 14
Product training . . . 14
Providing feedback on ABB Drives manuals . . . 14

Start-up

What this chapter contains . . . 15
How to start up the drive . . . 15
How to control the drive through the I/O interface . . . 27

Drive programming using PC tools

What this chapter contains . . . 29
General . . . 29
Programming via parameters . . . 30
Application programming . . . 30
- Function blocks . . . 31
- User parameters . . . 31
- Application events . . . 31
- Program execution . . . 31
- Solution program licensing and protection . . . 32
- Operation modes . . . 32

Drive control and features

What this chapter contains . . . 35
Local control vs. external control . . . 35
Operating modes of the drive . . . 36
- Speed control mode . . . 36
- Torque control mode . . . 36
- Drive control chain for speed and torque control . . . 37

Motor control features . . . 38
- Scalar motor control . . . 38
- Autophasing . . . 38
- Thermal motor protection . . . 39

DC voltage control features . . . 42
Overvoltage control . . . 42
Undervoltage control . . . 42
Voltage control and trip limits . . . 42
Braking chopper . . . 43
Low voltage mode . . . 43
Speed control features . . . 45
Jogging . . . 45
Speed controller tuning . . . 46
Motor feedback features . . . 48
Motor encoder gear function . . . 48
Mechanical brake control . . . 49
Emergency stop . . . 53
Miscellaneous features . . . 53
Backup and restore of drive contents . . . 53

Default connections of the control unit

What this chapter contains . . . 55

Parameters and firmware blocks

What this chapter contains . . . 57
Types of parameters . . . 57
Firmware blocks . . . 58
Group *01 ACTUAL VALUES* . . . 59
ACTUAL VALUES . . . 59
Group *02 I/O VALUES* . . . 62
Group *03 CONTROL VALUES* . . . 68
Group *06 DRIVE STATUS* . . . 70
Group *08 ALARMS & FAULTS* . . . 74
Group *09 SYSTEM INFO* . . . 77
Group *10 START/STOP* . . . 78
DRIVE LOGIC . . . 78
Group *11 START/STOP MODE* . . . 84
START/STOP MODE . . . 84
Group *12 DIGITAL IO* . . . 87
DIO1 . . . 87
DIO2 . . . 87
DIO3 . . . 87
RO . . . 89
DI . . . 89
Group *13 ANALOGUE INPUTS* . . . 91
AI1 . . . 91
AI2 . . . 92
Group *15 ANALOGUE OUTPUTS* . . . 95
AO1 . . . 95
AO2 . . . 96
Group *16 SYSTEM* . . . 98
Group *17 PANEL DISPLAY* . . . 101

Group *20 LIMITS* . . . 102
LIMITS . . . 102
Group *22 SPEED FEEDBACK* . . . 104
SPEED FEEDBACK . . . 105
Group *24 SPEED REF MOD* . . . 109
SPEED REF SEL . . . 110
SPEED REF MOD . . . 111
Group *25 SPEED REF RAMP* . . . 113
SPEED REF RAMP . . . 114
Group *26 SPEED ERROR* . . . 117
SPEED ERROR . . . 118
Group *28 SPEED CONTROL* . . . 121
SPEED CONTROL . . . 122
Group *32 TORQUE REFERENCE* . . . 128
TORQ REF SEL . . . 129
TORQ REF MOD . . . 130
Group *33 SUPERVISION* . . . 132
SUPERVISION . . . 132
Group *34 REFERENCE CTRL* . . . 135
REFERENCE CTRL . . . 136
Group *35 MECH BRAKE CTRL* . . . 139
MECH BRAKE CTRL . . . 139
Group *40 MOTOR CONTROL* . . . 142
MOTOR CONTROL . . . 142
Group *45 MOT THERM PROT* . . . 145
MOT THERM PROT . . . 145
Group *46 FAULT FUNCTIONS* . . . 149
FAULT FUNCTIONS . . . 149
Group *47 VOLTAGE CTRL* . . . 152
VOLTAGE CTRL . . . 152
Group *48 BRAKE CHOPPER* . . . 154
BRAKE CHOPPER . . . 154
Group *50 FIELDBUS* . . . 156
FIELDBUS . . . 156
Group *51 FBA SETTINGS* . . . 159
Group *52 FBA DATA IN* . . . 161
Group *53 FBA DATA OUT* . . . 162
Group *55 COMMUNICATION TOOL* . . . 163
Group *57 D2D COMMUNICATION* . . . 164
D2D COMMUNICATION . . . 164
Group *90 ENC MODULE SEL* . . . 168
ENCODER . . . 169
Group *91 ABSOL ENC CONF* . . . 173
ABSOL ENC CONF . . . 173
Group *92 RESOLVER CONF* . . . 178
RESOLVER CONF . . . 178
Group *93 PULSE ENC CONF* . . . 179
PULSE ENC CONF . . . 179
Group *95 HW CONFIGURATION* . . . 182
Group *97 USER MOTOR PAR* . . . 183

Group *98 MOTOR CALC VALUES* 186
Group *99 START-UP DATA* 187

Parameter data

What this chapter contains 193
Terms 193
Fieldbus equivalent 194
Fieldbus addresses 194
Pointer parameter format in fieldbus communication 194
32-bit integer value pointers 194
32-bit integer bit pointers 195
Actual signals (Parameter groups 1…9) 196

Fault tracing

What this chapter contains 209
Safety 209
Alarm and fault indications 209
How to reset 209
Fault history 210
Alarm messages generated by the drive 211
Fault messages generated by the drive 219

Standard function blocks

What this chapter contains 231
Terms 231
Alphabetical index 232
Arithmetic 233
ABS 233
ADD 233
DIV 233
EXPT 234
MOD 234
MOVE 235
MUL 235
MULDIV 235
SQRT 236
SUB 236
Bitstring 237
AND 237
NOT 237
OR 238
ROL 238
ROR 239
SHL 239
SHR 240
XOR 240

Bitwise . . . 242
BGET . . . 242
BITAND . . . 242
BITOR . . . 243
BSET . . . 243
REG . . . 244
SR-D . . . 245
Communication . . . 246
D2D_Conf . . . 246
D2D_McastToken . . . 247
D2D_SendMessage . . . 247
DS_ReadLocal . . . 249
DS_WriteLocal . . . 250
Comparison . . . 251
EQ . . . 251
GE . . . 251
GT . . . 251
LE . . . 252
LT . . . 252
NE . . . 253
Conversion . . . 254
BOOL_TO_DINT . . . 254
BOOL_TO_INT . . . 255
DINT_TO_BOOL . . . 256
DINT_TO_INT . . . 257
DINT_TO_REALn . . . 257
DINT_TO_REALn_SIMP . . . 258
INT_TO_BOOL . . . 259
INT_TO_DINT . . . 259
REAL_TO_REAL24 . . . 260
REAL24_TO_REAL . . . 260
REALn_TO_DINT . . . 261
REALn_TO_DINT_SIMP . . . 261
Counters . . . 263
CTD . . . 263
CTD_DINT . . . 263
CTU . . . 264
CTU_DINT . . . 265
CTUD . . . 266
CTUD_DINT . . . 268
Edge & bistable . . . 270
FTRIG . . . 270
RS . . . 270
RTRIG . . . 271
SR . . . 272
Extensions . . . 273
FIO_01_slot1 . . . 273
FIO_01_slot2 . . . 274
FIO_11_AI_slot1 . . . 275
FIO_11_AI_slot2 . . . 277

FIO_11_AO_slot1 . . . 279
FIO_11_AO_slot2 . . . 280
FIO_11_DIO_slot1 . . . 282
FIO_11_DIO_slot2 . . . 282
Feedback & algorithms . . . 284
CRITSPEED . . . 284
CYCLET . . . 285
DATA CONTAINER . . . 285
FUNG-1V . . . 286
INT . . . 287
MOTPOT . . . 288
PID . . . 289
RAMP . . . 291
REG-G . . . 292
SOLUTION_FAULT . . . 294
Filters . . . 295
FILT1 . . . 295
FILT2 . . . 295
LEAD/LAG . . . 297
Parameters . . . 298
GetBitPtr . . . 298
GetValPtr . . . 298
PARRD . . . 298
PARRDINTR . . . 299
PARRDPTR . . . 299
PARWR . . . 300
Program structure . . . 301
BOP . . . 301
ELSE . . . 301
ELSEIF . . . 301
ENDIF . . . 302
IF . . . 302
Selection . . . 304
LIMIT . . . 304
MAX . . . 304
MIN . . . 304
MUX . . . 305
SEL . . . 305
Switch & Demux . . . 306
DEMUX-I . . . 306
DEMUX-MI . . . 306
SWITCH . . . 307
SWITCHC . . . 308
Timers . . . 309
MONO . . . 309
TOF . . . 310
TON . . . 310
TP . . . 311

Application program template

What this chapter contains . . . 313

Appendix A – Fieldbus control

What this chapter contains . . . 325
System overview . . . 325
Setting up communication through a fieldbus adapter module . . . 326
Drive control parameters . . . 327
The fieldbus control interface . . . 328
The Control Word and the Status Word . . . 328
Actual values . . . 329
FBA communication profile . . . 329
Fieldbus references . . . 329
State diagram . . . 330

Appendix B – Drive-to-drive link

What this chapter contains . . . 331
General . . . 331
Wiring . . . 331
Datasets . . . 332
Types of messaging . . . 333
Master point-to-point messaging . . . 334
Read remote messaging . . . 334
Follower point-to-point messaging . . . 335
Standard multicast messaging . . . 335
Broadcast messaging . . . 336
Chained multicast messaging . . . 337
Examples of using standard function blocks in drive-to-drive communication . . . 339
Example of master point-to-point messaging . . . 339
Example of read remote messaging . . . 340
Releasing tokens for follower-to-follower communication . . . 341
Example of follower point-to-point messaging . . . 341
Example of standard master-to-follower(s) multicast messaging . . . 342
Example of broadcast messaging . . . 342

Appendix C – Control chain and drive logic diagrams

What this chapter contains . . . 343

Introduction to the manual

What this chapter contains

The chapter includes a description of the contents of the manual. In addition it contains information about the compatibility, safety and intended audience.

Compatibility

The manual is compatible with ACSM1 Speed and Torque Control program version UMFI1510 and later. See parameter 9.04 FIRMWARE VER or PC tool (View - Properties).

Safety instructions

Follow all safety instructions delivered with the drive.

- Read the **complete safety instructions** before you install, commission, or use the drive. The complete safety instructions are given at the beginning of the *Hardware Manual*.
- Read the **software function specific warnings and notes** before changing the default settings of the function. For each function, the warnings and notes are given in this manual in the section describing the related user-adjustable parameters.

Reader

The reader of the manual is expected to know the standard electrical wiring practices, electronic components, and electrical schematic symbols.

Contents

The manual consists of the following chapters:

- *Start-up* instructs in setting up the control program and how to control the drive through the I/O interface.
- *Drive programming using PC tools* introduces programming via PC tool (DriveStudio and/or DriveSPC).
- *Drive control and features* describes the control locations and operation modes of the drive, and the features of the application program.
- *Default connections of the control unit* presents the default connections of the JCU Control Unit.
- *Parameters and firmware blocks* describes the drive parameters and firmware function blocks.
- *Parameter data* contains more information on the parameters of the drive.
- *Fault tracing* lists the warning and fault messages with the possible causes and remedies.
- *Standard function blocks*
- *Application program template*
- *Appendix A – Fieldbus control* describes the communication between the drive and a fieldbus.
- *Appendix B – Drive-to-drive link* describes the communication between drives connected together by the drive-to-drive link.
- *Appendix C – Control chain and drive logic diagrams*.

Product and service inquiries

Address any inquiries about the product to your local ABB representative, quoting the type code and serial number of the unit in question. A listing of ABB sales, support and service contacts can be found by navigating to www.abb.com/drives and selecting *Drives – Sales, Support and Service network*.

Product training

For information on ABB product training, navigate to www.abb.com/drives and select *Drives – Training courses*.

Providing feedback on ABB Drives manuals

Your comments on our manuals are welcome. Go to www.abb.com/drives and select *Document Library – Manuals feedback form (LV AC drives)*.

Start-up

What this chapter contains

This chapter describes the basic start-up procedure of the drive and instructs in how to control the drive through the I/O interface.

How to start up the drive

The drive can be operated:

- locally from PC tool or control panel
- externally via I/O connections or fieldbus interface.

The start-up procedure presented uses the DriveStudio PC tool program. Drive references and signals can be monitored with DriveStudio (Data Logger or Monitor Window). For instructions on how to use DriveStudio, see *DriveStudio User Manual* [3AFE68749026 (English)].

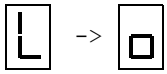
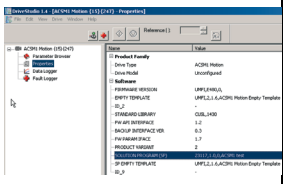
The start-up procedure includes actions which need to be performed only when the drive is powered up for the first time (e.g. entering the motor data). After the first start-up, the drive can be powered up without using these start-up functions. The start-up procedure can be repeated later if start-up data needs to be changed.

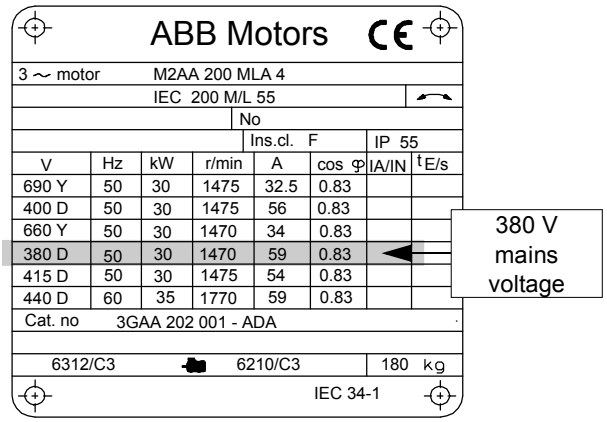
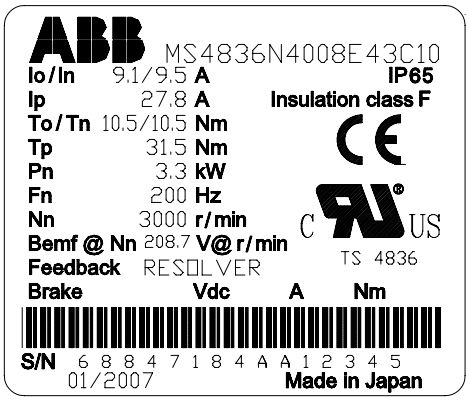
In addition to the PC tool commissioning and drive power-up, the start-up procedure includes the following steps:

- entering the motor data and performing the motor identification run
- setting up the encoder/resolver communication
- checking the emergency stop and Safe Torque Off circuits
- setting up the voltage control
- setting the drive limits
- setting up the motor overtemperature protection
- tuning the speed controller
- setting up the fieldbus control.

If an alarm or a fault is generated during the start-up, see chapter *Fault tracing* for the possible causes and remedies. If problems continue, disconnect the main power and wait 5 minutes for the intermediate circuit capacitors to discharge and check the drive and motor connections.

Before you start, ensure you have the motor nameplate and encoder data (if needed) at your hand.

Safety		
	The start-up may only be carried out by a qualified electrician. The safety instructions must be followed during the start-up procedure. See the safety instructions on the first pages of the appropriate hardware manual.	
☐	Check the installation. See the installation checklist in the appropriate hardware manual.	
☐	Check that the starting of the motor does not cause any danger. **De-couple the driven machine** if - there is a risk of damage in case of an incorrect direction of rotation, or - a normal ID run (99.13 IDRUN MODE = (1) Normal) is required during the drive start-up, when the load torque is higher than 20% or the machinery is not able to withstand the nominal torque transient during the ID run.	
PC tool		
☐	Install the DriveStudio PC tool onto the PC. Install also DriveSPC if block programming is needed. For instructions, see *DriveStudio User Manual* [3AFE68749026 (English)] and *DriveSPC User Manual* [3AFE68836590 (English)].	
☐	Connect the drive to the PC: Connect the other end of the communication cable (OPCA-02, code: 68239745) to the panel link of the drive. Connect the other end of the communication cable via USB adapter or directly to the PC serial port.	
Power up		
☐	Switch the power on.	7-segment display: 
Note: The drive will indicate an alarm (2021 NO MOTOR DATA) until the motor data is entered later in this procedure. This is completely normal.		
☐	Start the DriveStudio program by clicking the DriveStudio icon on the PC desktop.	
☐	Check whether an application program exists using the DriveStudio tool. If it does, the rows SOLUTION PROGRAM (SP) and SP EMPTY TEMPLATE are displayed in drive properties (View - Properties, Software category). If an application program already exists, NOTE that some of the drive functions may have been disabled. ENSURE that the application program is suitable for your drive application.	
☐	Switch to local control to ensure that external control is disabled by clicking the Take/Release button of the DriveStudio tool control panel.	

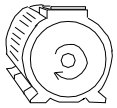
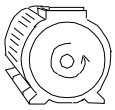
Motor data entering		
☐	Open the parameter and signal list by selecting the Parameter Browser of the appropriate drive.	Parameter Browser
☐	Select the language. Parameters are set as follows: Select the parameter group (in this case 99 START-UP DATA) by double-clicking it. Select the appropriate parameter by double-clicking it and set the new value.	99.01 LANGUAGE
☐	Select the motor type: asynchronous or permanent magnet motor.	99.04 MOTOR TYPE
☐	Select the motor control mode. DTC is suitable for most cases. For information on scalar control, see parameter 99.05 MOTOR CTRL MODE.	99.05 MOTOR CTRL MODE
☐	Enter the motor data from the motor nameplate. Asynchronous motor nameplate example:  Permanent magnet motor nameplate example:  With DTC control (99.05 MOTOR CTRL MODE = (0) DTC) at least parameters 99.06…99.10 must be set. Better control accuracy can be achieved by setting also parameters 99.11…99.12.	**Note:** Set the motor data to exactly the same value as on the motor nameplate. For example, if the motor nominal speed is 1470 rpm on the nameplate, setting the value of parameter 99.09 MOT NOM SPEED to 1500 rpm results in wrong operation of the drive.

efesotomasyon.com

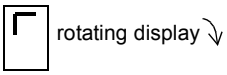
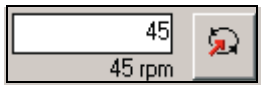
	- motor nominal current Allowed range: approximately 1/6 · I_{2n} … 2 · I_{2n} of the drive (0…2 · I_{2nd} if parameter 99.05 MOTOR CTRL MODE = (1) Scalar). With multimotor drives, see section *Multimotor drives* on page 19.	99.06 MOT NOM CURRENT
	- motor nominal voltage Allowed range: 1/6 · U_N … 2 · U_N of the drive. (U_N refers to the highest voltage in each of the nominal voltage range, i.e. 480 V AC for ACSM1-04). With permanent magnet motors: The nominal voltage is the BackEMF voltage (at motor nominal speed). If the voltage is given as voltage per rpm, e.g. 60 V per 1000 rpm, the voltage for 3000 rpm nominal speed is 3 × 60 V = 180 V. Note that the nominal voltage is not equal to the equivalent DC motor voltage (E.D.C.M.) value given by some motor manufactures. The nominal voltage can be calculated by dividing the E.D.C.M. voltage by 1.7 (= square root of 3).	99.07 MOT NOM VOLTAGE
	- motor nominal frequency Range: 5…500 Hz. With multimotor drives, see section *Multimotor drives* on page 19. With permanent magnet motor: If the frequency is not given on the motor nameplate, it has to be calculated with the following formula: $f = n \times p / 60$ where p = number of pole pairs, n = motor nominal speed.	99.08 MOT NOM FREQ
	- motor nominal speed Range: 0…10000 rpm. With multimotor drives, see section *Multimotor drives* on page 19.	99.09 MOT NOM SPEED
	- motor nominal power Range: 0…10000 kW. With multimotor drives, see section *Multimotor drives* on page 19.	99.10 MOT NOM POWER
	- motor nominal cosφ (not applicable for permanent magnet motors). This value can be set for better DTC control accuracy. If value is not given by the motor manufacturer, use value 0 (i.e. default value). Range: 0…1.	99.11 MOT NOM COSFII
	- motor nominal shaft torque. This value can be set for better DTC control accuracy. If value is not given by the motor manufacturer, use value 0 (i.e. default value). Range: 0…2147483.647 Nm.	99.12 MOT NOM TORQUE
☐	After the motor parameters have been set, alarm ID-RUN is generated to inform that the ID run needs to be performed.	Alarm: ID-RUN

Multimotor drives **I.e. more than one motor is connected to one drive.**		
☐	Check that the motors have the same relative slip (only for asynchronous motors), nominal voltage and number of poles. If the manufacturer motor data is insufficient, use the following formulas to calculate the slip and the number of poles: $p = \mathrm{Int}\left(\frac{f_N \cdot 60}{n_N}\right)$ $n_s = \frac{f_N \cdot 60}{p}$ $s = \frac{n_S - n_N}{n_S} \cdot 100\%$ where p = number of pole pairs (= motor pole number / 2) f_N = motor nominal frequency [Hz] n_N = motor nominal speed [rpm] s = motor slip [%] n_S = motor synchronous speed [rpm].	
☐	Set the sum of the motor nominal currents.	99.06 MOT NOM CURRENT
☐	Set the nominal motor frequencies. Frequencies must be the same.	99.08 MOT NOM FREQ
☐	Set the sum of the motor nominal powers. If the motor powers are close to each other or the same but the nominal speeds vary slightly, parameter 99.09 MOT NOM SPEED can be set to an average value of the motor speeds.	99.10 MOT NOM POWER 99.09 MOT NOM SPEED
External control unit power supply		
☐	If the control unit of the drive is powered from an external power supply (as specified in *Hardware Manual)*, set parameter 95.01 CTRL UNIT SUPPLY to EXTERNAL 24V.	95.01 CTRL UNIT SUPPLY
External mains choke		
☐	If the drive is equipped with an external choke (specified in *Hardware Manual)*, set parameter 95.02 EXTERNAL CHOKE to YES.	95.02 EXTERNAL CHOKE
Motor overtemperature protection (1)		
☐	Select how the drive reacts when motor overtemperature is detected.	45.01 MOT TEMP PROT
☐	Select the motor temperature protection: motor thermal model or motor temperature measurement. For motor temperature measurement connections, see section *Temperature sensors* on page 40.	45.02 MOT TEMP SOURCE

efesotomasyon.com

ID RUN (motor identification run)		
	WARNING! With Normal or Reduced ID run the motor will run at up to approximately 50…100% of the nominal speed during the ID run. ENSURE THAT IT IS SAFE TO RUN THE MOTOR BEFORE PERFORMING THE ID RUN!	
Note: Ensure that possible Safe Torque Off and emergency stop circuits are closed during the ID run.		
☐	Check the direction of rotation of the motor before starting the ID run. During the run (Normal or Reduced), the motor will rotate in the forward direction.	When drive output phases U2, V2 and W2 are connected to the corresponding motor terminals:  forward direction  reverse direction

☐	Select the motor identification method by parameter 99.13 IDRUN MODE. During the Motor ID run, the drive will identify the characteristics of the motor for optimum motor control. The ID run is performed at the next start of the drive. **Note:** The motor shaft must NOT be locked and the load torque must be < 20% during Normal or Reduced ID run. With permanent magnet motor this restriction applies also when Standstill ID run is selected. **Note:** Mechanical brake (if present) is not opened during the ID run. **Note:** The ID run cannot be performed if par. 99.05 MOTOR CTRL MODE = (1) Scalar. NORMAL ID run should be selected whenever possible. **Note:** The driven machinery must be de-coupled from the motor with Normal ID run if • the load torque is higher than 20%, or • the machinery is not able to withstand the nominal torque transient during the ID run. The REDUCED ID run should be selected instead of the Normal ID run if the mechanical losses are higher than 20%, i.e. the motor cannot be de-coupled from the driven equipment, or full flux is required to keep the motor brake open (conical motor). The STANDSTILL ID run should be selected only if the Normal or Reduced ID run is not possible due to the restrictions caused by the connected mechanics (e.g. with lift or crane applications). AUTOPHASING can only be selected after the Normal/Reduced/ Standstill ID run has been performed once. Autophasing is used when an absolute encoder or a resolver (or encoder with commutation signals) has been added/changed to a permanent magnet motor, but there is no need to perform the Normal/Reduced/Standstill ID run again. See parameter 11.07 AUTOPHASING MODE on page 86 for information on autophasing modes, and section *Autophasing* on page 38.	99.13 IDRUN MODE 11.07 AUTOPHASING MODE
☐	Check the drive limits. The following must apply for all ID run methods: • 20.05 MAXIMUM CURRENT > 99.06 MOT NOM CURRENT In addition, the following must apply for Reduced and Normal ID run: • 20.01 MAXIMUM SPEED > 50% of synchronous speed of the motor • 20.02 MINIMUM SPEED ≤ 0 • supply voltage ≥ 66% of 99.07 MOT NOM VOLTAGE • 20.06 MAXIMUM TORQUE ≥ 100% (asynchronous machines with Normal ID run only) • 20.06 MAXIMUM TORQUE ≥ 30% (asynchronous machines with Reduced ID run, and permanent magnet motors). When the ID run has been successfully completed, set the limit values as required by the application.	

☐	Start the motor to activate the ID run. **Note:** RUN ENABLE must be active. ID run is indicated by alarm ID-RUN and by a rotating display on the 7-segment display.	 10.09 RUN ENABLE Alarm: ID-RUN 7-segment display: 
☐	If the ID run is not successfully completed, fault ID-RUN FAULT is generated.	Fault ID-RUN FAULT
Speed measurement with encoder/resolver		
An encoder/resolver feedback can be used for more accurate motor control. Follow these instructions when encoder/resolver interface module FEN-xx is installed in drive option Slot 1 or 2. **Note:** Two encoder interface modules of the same type are not allowed.		
☐	Select the used encoder/resolver. For more information, see parameter group 90 ENC MODULE SEL on page 169.	90.01 ENCODER 1 SEL / 90.02 ENCODER 2 SEL
☐	Set other necessary encoder/resolver parameters: - Absolute encoder parameters (group 91, page 173) - Resolver parameters (group 92, page 178). - Pulse encoder parameters (group 93, page 179).	91.01…91.31 / 92.01…92.03 / 93.01…93.22
☐	Set parameter 90.10 ENC PAR REFRESH to (1) Configure so that the new parameter settings take effect.	90.10 ENC PAR REFRESH
Checking the encoder/resolver connection		
Follow these instructions when encoder/resolver interface module FEN-xx is installed in drive option Slot 1 or 2. **Note:** Two encoder interface modules of the same type are not allowed.		
☐	Set parameter 22.01 SPEED FB SEL to (0) Estimated.	22.01 SPEED FB SEL
☐	Enter a small speed reference value (for example 3% of the nominal motor speed).	
☐	Start the motor.	
☐	Check that the estimated (1.14 SPEED ESTIMATED) and actual speed (1.08 ENCODER 1 SPEED / 1.10 ENCODER 2 SPEED) are equal. If the values differ, check the encoder/resolver parameter settings. **Hint:** If the actual speed (with absolute or pulse encoder) differs form the reference value by a factor of 2, check the pulse number setting (91.01 SINE COSINE NR / 93.01 ENC1 PULSE NR / 93.11 ENC2 PULSE NR).	1.14 SPEED ESTIMATED 1.08 ENCODER 1 SPEED / 1.10 ENCODER 2 SPEED

☐	If the direction of rotation is selected as forward, check that the actual speed (1.08 ENCODER 1 SPEED / 1.10 ENCODER 2 SPEED) is positive: • If the actual direction of rotation is forward and the actual speed negative, the phasing of the pulse encoder wires is reversed. • If the actual direction of rotation is reverse and the actual speed negative, the motor cables are incorrectly connected. Changing the connection: Disconnect the main power, and wait for 5 minutes for the intermediate circuit capacitors to discharge. Do the necessary changes. Switch the power on and start the motor again. Check that the estimated and actual speed values are correct. • If the direction of rotation is selected as reverse, the actual speed must be negative. **Note:** Resolver autotuning routines should always be performed after resolver cable connection has been modified. Autotuning routines can be activated by setting parameter 92.02 EXC SIGNAL AMPL or 92.03 EXC SIGNAL FREQ, and then setting parameter 90.10 ENC PAR REFRESH to (1) Configure. If the resolver is used with a permanent magnet motor, an AUTOPHASING ID run should be performed as well.	1.08 ENCODER 1 SPEED / 1.10 ENCODER 2 SPEED
☐	Stop the motor.	
☐	Set parameter 22.01 SPEED FB SEL to (1) Enc1 speed or (2) Enc2 speed. If the speed feedback cannot be used in motor control: In special applications parameter 40.06 FORCE OPEN LOOP must be set to TRUE.	22.01 SPEED FB SEL
☐	**Note:** Speed filtering needs to be adjusted especially when the encoder pulse number is small. See section *Speed filtering* on page 25.	
Emergency stop circuit		
☐	If there is an emergency stop circuit in use, check that the circuit functions (emergency stop signal is connected to the digital input which is selected as the source for the emergency stop activation).	10.10 EM STOP OFF3 or 10.11 EM STOP OFF1 (emergency stop control through fieldbus 2.12 FBA MAIN CW bits 2…4)

Safe Torque Off		
The Safe Torque Off function disables the control voltage of the power semiconductors of the drive output stage, thus preventing the inverter from generating the voltage required to rotate the motor. For Safe Torque Off wiring, see the appropriate hardware manual.		
☐	If there is a Safe Torque Off circuit in use, check that the circuit functions.	
☐	Selects how the drive reacts when the Safe Torque Off function is active (i.e. when the control voltage of the power semiconductors of the drive output stage is disabled).	46.07 STO DIAGNOSTIC
Voltage control		
If the DC voltage drops due to input power cut off, the undervoltage controller will automatically decrease the motor torque in order to keep the voltage above the lower limit. To prevent the DC voltage from exceeding the overvoltage control limit, the overvoltage controller automatically decreases the generating torque when the limit is reached. When the overvoltage controller is limiting the generating torque, quick deceleration of the motor is not possible. Thus electrical braking (brake chopper and brake resistor) is needed in some applications to allow the drive to dissipate regenerative energy. The chopper connects the brake resistor to the intermediate circuit of the drive whenever the DC voltage exceeds the maximum limit.		
☐	Check that the overvoltage and undervoltage controllers are active.	47.01 OVERVOLTAGE CTRL 47.02 UNDERVOLT CTRL
☐	If the application requires a brake resistor (the drive has a built-in brake chopper): • Set the brake chopper and resistor settings. **Note:** When a brake chopper and resistor are used, the overvoltage controller must be deactivated by parameter 47.01 OVERVOLTAGE CTRL. • Check that the connection functions. For more information on the brake resistor connection, see the appropriate hardware manual.	48.01…48.07 47.01 OVERVOLTAGE CTRL
Start function		
☐	Select the start function. Setting 11.01 START MODE to (2) Automatic selects a general-purpose start function. This setting also makes flying start (starting to a rotating motor) possible. The highest possible starting torque is achieved when 11.01 START MODE is set to (0) Fast (automatic optimised DC magnetising) or (1) Const time (constant DC magnetising with user-defined magnetising time). **Note:** When 11.01 START MODE setting is (0) Fast or (1) Const time, flying start (start to a rotating motor) is not possible.	11.01 START MODE

Limits		
☐	Set the operation limits according to the process requirements. **Note:** If load torque is suddenly lost when the drive is operating in torque control mode, the drive will rush to the defined negative or positive maximum speed. For safe operation, ensure the set limits are suitable for your application.	20.01…20.07
Motor overtemperature protection (2)		
☐	Set the alarm and fault limits for the motor overtemperature protection.	45.03 MOT TEMP ALM LIM 45.04 MOT TEMP FLT LIM
☐	Set the typical ambient temperature of the motor.	45.05 AMBIENT TEMP
☐	When 45.02 MOT TEMP SOURCE is set to (0) ESTIMATED, the motor thermal protection model must be configured as follows: - Set the maximum allowed operating load of the motor. - Set the zero speed load. A higher value can be used if the motor has an external motor fan to boost the cooling. - Set the break point frequency of the motor load curve. - Set the motor nominal temperature rise. - Set the time inside which the temperature has reached 63% of the nominal temperature.	45.06 MOT LOAD CURVE 45.07 ZERO SPEED LOAD 45.08 BREAK POINT 45.09 MOTNOM TEMP RISE 45.10 MOT THERM TIME
☐	If possible, perform the motor ID run again at this point (see page 20).	99.13 IDRUN MODE
Speed filtering		
The measured speed always has a small ripple because of electrical and mechanical interferences, couplings and encoder resolution (i.e. small pulse number). A small ripple is acceptable as long as it does not affect the speed control chain. The interferences in the speed measurement can be filtered with a speed error filter or with an actual speed filter. Reducing the ripple with filters may cause speed controller tuning problems. A long filter time constant and fast acceleration time contradict one another. A very long filter time results in unstable control.		
☐	If the used speed reference changes rapidly (servo application), use the speed error filter to filter the possible interferences in the speed measurement. In this case the speed error filter is more suitable than the actual speed filter: - Set the filter time constant.	26.06 SPD ERR FTIME

☐	If the used speed reference remains constant, use the actual speed filter to filter the possible interferences in the speed measurement. In this case the actual speed filter is more suitable than the speed error filter: - Set the filter time constant. If there are substantial interferences in the speed measurement, the filter time constant should be proportional to the total inertia of the load and motor, i.e. approximately 10…30% of the mechanical time constant $t_{mech} = (n_{nom} / T_{nom}) \times J_{tot} \times 2\pi / 60$, where J_{tot} = total inertia of the load and motor (the gear ratio between the load and the motor must be taken into account) n_{nom} = motor nominal speed T_{nom} = motor nominal torque	22.02 SPEED ACT FTIME
Speed controller tuning		
For the most demanding applications, the P- and I-parts of the speed controller of the drive can be tuned either manually or automatically. See parameter 28.16 PI TUNE MODE. If it is necessary to adjust acceleration (deceleration) compensation, it must be done manually.		
☐	Acceleration (deceleration) compensation can be used to improve the speed control dynamic reference change (when the speed ramp times > 0). In order to compensate inertia during acceleration, a derivative of the speed reference is added to the output of the speed controller. Set the derivation time for acceleration (deceleration) compensation. The value should be proportional to the total inertia of the load and motor, i.e. approximately 50…100% of the mechanical time constant (t_{mech}). See the mechanical time constant equation in section *Speed filtering* on page 25.	26.08 ACC COMP DERTIME
Fieldbus control		
Follow these instructions when the drive is controlled from a fieldbus control system via fieldbus adapter Fxxx. The adapter is installed in drive Slot 3.		
☐	Enable the communication between the drive and fieldbus adapter.	50.01 FBA ENABLE
☐	Connect the fieldbus control system to the fieldbus adapter module.	
☐	Set the communication and adapter module parameters: See section *Setting up communication through a fieldbus adapter module* on page 326.	
☐	Test that the communication functions.	

How to control the drive through the I/O interface

The table below instructs how to operate the drive through the digital and analogue inputs, when the default parameter settings are valid.

PRELIMINARY SETTINGS	
Ensure the control connections are wired according to the connection diagram given in chapter *Default connections of the control unit*.	
Switch to external control by clicking the Take/Release button of the PC tool control panel.	
STARTING AND CONTROLLING THE SPEED OF THE MOTOR	
Start the drive by switching digital input DI1 on. Digital input status can be monitored with signal 2.01 DI STATUS.	2.01 DI STATUS
Check that analogue input AI1 is used as a voltage input (selected by jumper J1).	Voltage: J1
Regulate the speed by adjusting the voltage of analogue input AI1.	
Check analogue input AI1 signal scaling. AI1 values can be monitored with signals 2.04 AI1 and 2.05 AI1 SCALED. When AI1 is used as a voltage input, the input is differential and the negative value corresponds to the negative speed and the positive value to the positive speed.	13.02…13.04 2.04 AI1 2.05 AI1 SCALED
STOPPING THE MOTOR	
Stop the drive by switching digital input DI1 off.	2.01 DI STATUS

Drive programming using PC tools

What this chapter contains

This chapter introduces the drive programming using the DriveStudio and DriveSPC applications. For more information, see *DriveStudio User Manual* [3AFE68749026 (English)] and *DriveSPC User Manual* [3AFE68836590 (English)].

General

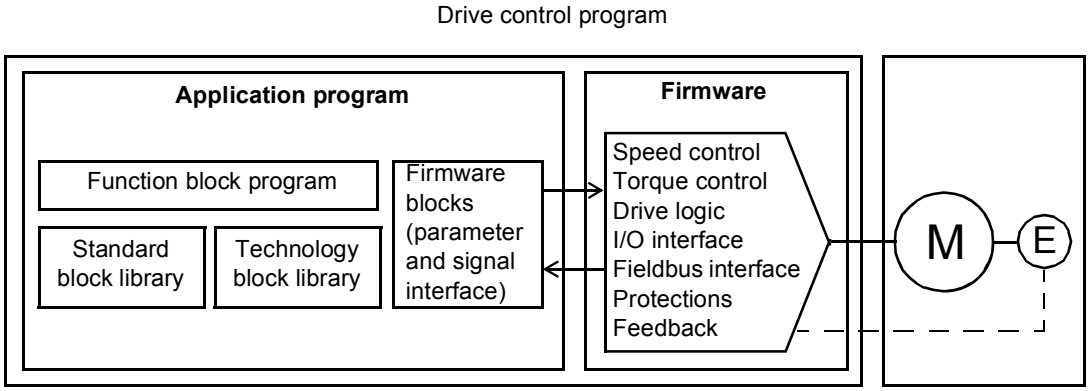
The drive control program is divided into two parts:

- firmware program
- application program.

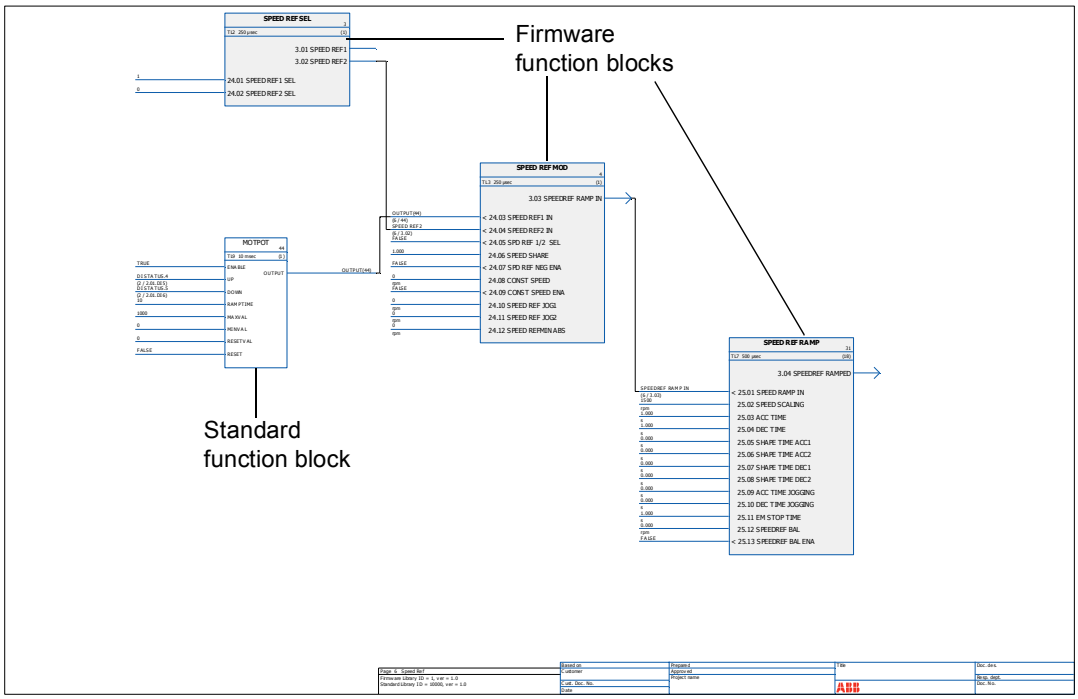
The firmware program performs the main control functions, including speed and torque control, drive logic (start/stop), I/O, feedback, communication and protection functions. Firmware functions are configured and programmed with parameters. The functions of the firmware program can be extended with application programming. Application programs are built out of function blocks.

The drive supports two different programming methods:

- parameter programming
- application programming with function blocks (the blocks are based on the IEC-61131 standard).



The following picture presents a view from DriveSPC.



The application program template visible through DriveSPC is presented in chapter *Application program template* (page 313).

Programming via parameters

Parameters can be set via DriveStudio, drive control panel (keypad) or the fieldbus interface. All parameter settings are stored automatically to the permanent memory of the drive. However, it is highly recommended to force a save by using parameter 16.07 PARAM SAVE before powering down the drive immediately after any parameter changes. Values are restored after the power switch-off. If necessary, the default values can be restored by parameter 16.04 PARAM RESTORE.

Because most parameters are used as firmware function block inputs, parameter values can also be modified via the application program.

Application programming

Application programs are created with the DriveSPC PC tool.

The normal delivery of the drive does not include an application program. The user can create an application program with the standard and firmware function blocks. ABB also offers customised application programs and technology function blocks for specific applications. For more information, contact your local ABB representative.

Function blocks

The application program uses three types of function blocks: firmware function blocks, standard function blocks and technology function blocks.

Firmware function blocks

Most of the firmware functions are represented as function blocks in the DriveSPC tool. Firmware function blocks are part of the drive control firmware, and used as an interface between the application and firmware programs. Drive parameters in groups 10…99 are used as function block inputs and parameters in groups 1…9 as function block outputs. Firmware function blocks are presented in chapter *Parameters and firmware blocks*.

Standard function blocks (library)

Standard function blocks (e.g. ADD, AND) are used to create an executable application program. The standard function blocks available are presented in chapter *Standard function blocks*.

Standard function block library is always included in the drive delivery.

Technology function blocks

Several technology function block libraries (e.g. CAM) are available for different types of applications. One technology library can be used at a time. Technology blocks are used in a similar way as the standard blocks.

User parameters

User parameters can be created with the DriveSPC tool. They are inserted into the application program as blocks that can be connected to existing application blocks.

User parameters can be added to any existing parameter group; the first available index is 70. Parameter groups 5 and 75…89 are available for user parameters starting from index 1. Using attributes, the parameters can be defined as write-protected, hidden, etc.

For more information, see the *DriveSPC User Manual.*

Application events

The application programmer can create his/her own application events (alarms and faults) by adding alarm and fault blocks; these blocks are managed through the Alarm and Fault Managers of the DriveSPC tool.

The operation of alarm and fault blocks is the same: when the block is enabled (by setting the Enable input to 1), an alarm or fault is generated by the drive.

Program execution

The application program is loaded to the permanent (non-volatile) memory of the memory unit (JMU). When the loading finishes, the drive control board is automatically reset, and the downloaded program started. The program is executed in real time on the same Central Processing Unit (CPU of the drive control board) as the drive firmware. The program can be executed at the two dedicated time levels of 1 and 10 milliseconds, as well as other time levels between certain firmware tasks.

Note: Because the firmware and application programs use the same CPU, the programmer must ensure that the drive CPU is not overloaded. See parameter 1.21 CPU USAGE.

Solution program licensing and protection

Note: This functionality is only available with DriveSPC version 1.5 and later.

The drive can be assigned an application licence consisting of an ID and password using the DriveSPC tool. Likewise, the application program created in DriveSPC can be protected by an ID and password. For instructions, refer to the DriveSPC user manual.

If a protected application program is downloaded to a licensed drive, the IDs and passwords of the application and drive must match. A protected application cannot be downloaded to an unlicensed drive. On the other hand, an unprotected application can be downloaded to a licensed drive.

The ID of the application licence is displayed by DriveStudio in the drive software properties as APPL LICENCE. If the value is 0, no licence has been assigned to the drive.

Notes:

- The application licence can only be assigned to a complete drive, not a stand-alone control unit.
- A protected application can only be downloaded to a complete drive, not a stand-alone control unit.

Operation modes

The DriveSPC tool offers the following operation modes:

Off-line

When the off-line mode is used without a drive connection, the user can

- open a application program file (if exists).
- modify and save the application program.
- print the program pages.

When the off-line mode is used with a drive(s) connection, the user can

- connect the selected drive to DriveSPC.
- upload an application program from the connected drive (an empty template which includes only the firmware blocks is available by default.)
- download the configured application program to the drive and start the program execution. The downloaded program contains the function block program and the parameter values set in DriveSPC.
- remove the program from the connected drive.

On-line

In the on-line mode, the user can

- modify firmware parameters (changes are stored directly to the drive memory)
- modify application program parameters (i.e. parameters created in DriveSPC)
- monitor the actual values of all function blocks in real time.

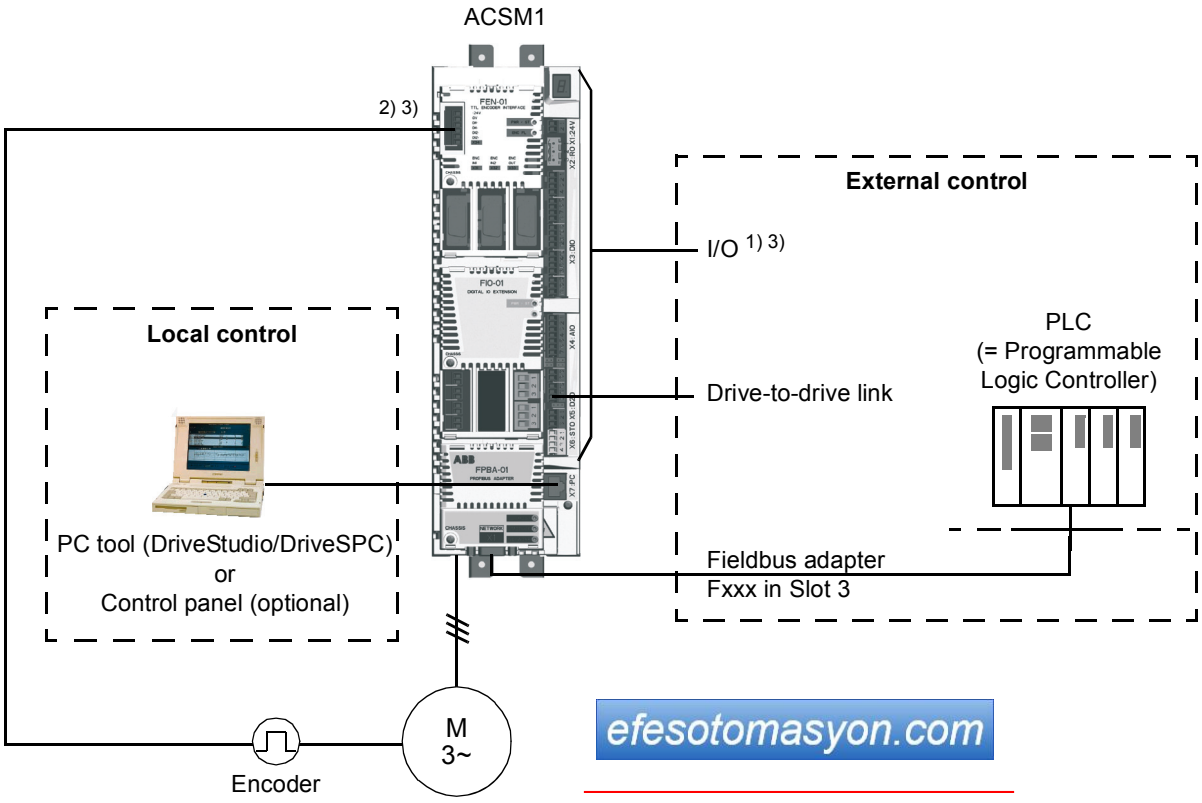
Drive control and features

What this chapter contains

This chapter describes the control locations and operation modes of the drive, and the features of the application program.

Local control vs. external control

The drive has two main control locations: external and local. The control location is selected with the PC tool (Take/Release button) or with the LOC/REM key on the control panel.



1) Extra inputs/outputs can be added by installing optional I/O extension modules (FIO-xx) in drive Slot 1/2.
2) Incremental or absolute encoder, or resolver interface module (FEN-xx) installed in drive Slot 1/2
3) Two encoder/resolver interface modules of the same type are not allowed.

Local control

The control commands are given from a PC equipped with DriveStudio and/or DriveSPC, or from the control panel keypad when the drive is in local control. Speed, torque and position control modes are available for local control.

Local control is mainly used during commissioning and maintenance. The control panel always overrides the external control signal sources when used in local control. Changing the control location to local can be disabled by parameter 16.01 LOCAL LOCK.

The user can select by a parameter (46.03 LOCAL CTRL LOSS) how the drive reacts to a control panel or PC tool communication break.

External control

When the drive is in external control, control commands (start/stop, reset etc.) are given through the fieldbus interface (via an optional fieldbus adapter module), the I/O terminals (digital inputs), optional I/O extension modules or the drive-to-drive link.

Two external control locations, EXT1 and EXT2, are available. The user can select control signals (e.g. start/stop and reference) and control modes for both external control locations. Depending on the user selection, either EXT1 or EXT2 is active at a time. Selection between EXT1/EXT2 is done via digital inputs or fieldbus control word. In addition, the EXT1 control location is divided into EXT1 CTRL MODE1 and EXT1 CTRL MODE2. Both use the EXT1 control signals for start/stop, but the control mode can be different; for example, EXT1 CTRL MODE2 can be used in homing.

Operating modes of the drive

The drive can operate in speed and torque control modes. A block diagram of the drive control chain for these modes is presented on page 37; more detailed diagrams are presented in *Appendix C – Control chain and drive logic diagrams* (page 343).

Speed control mode

Motor rotates at a speed proportional to the speed reference given to the drive. This mode can be used either with estimated speed used as feedback, or with an encoder or resolver for better speed accuracy.

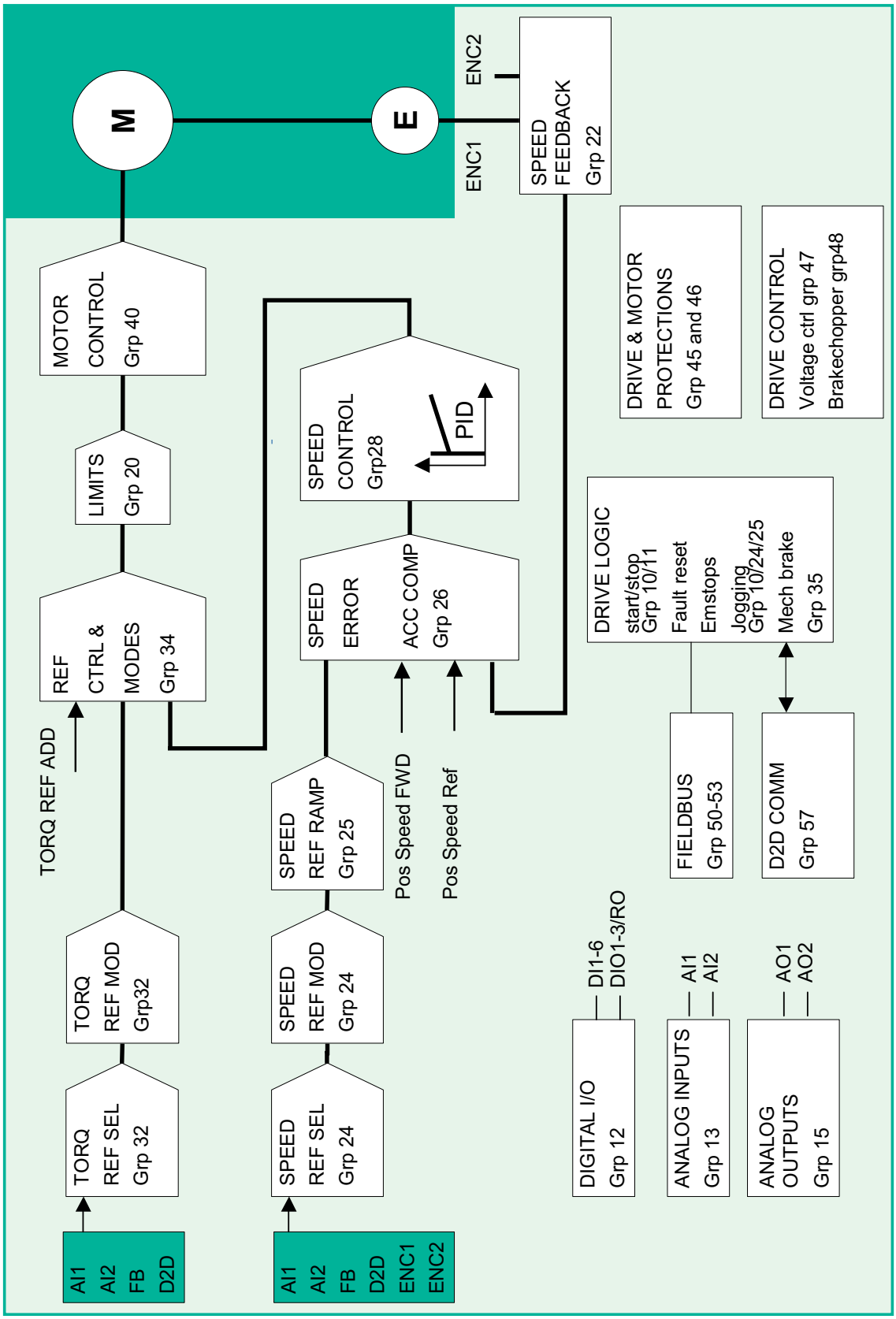
Speed control mode is available in both local and external control.

Torque control mode

Motor torque is proportional to the torque reference given to the drive. This mode can be used either with estimated speed used as feedback, or with an encoder or resolver for more accurate and dynamic motor control.

Torque control mode is available in both local and external control.

Drive control chain for speed and torque control



Motor control features

Scalar motor control

It is possible to select scalar control as the motor control method instead of Direct Torque Control (DTC). In scalar control mode, the drive is controlled with a frequency reference. However, the performance of DTC is not achieved in scalar control.

It is recommended to activate the scalar motor control mode in the following situations:

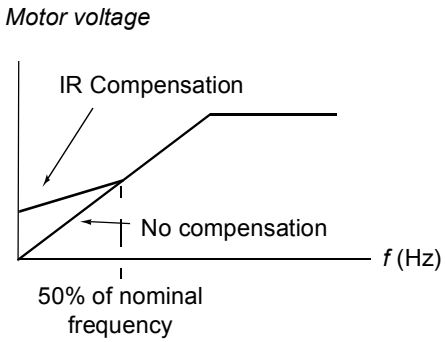
- In multimotor drives: 1) if the load is not equally shared between the motors, 2) if the motors are of different sizes, or 3) if the motors are going to be changed after motor identification (ID run)
- If the nominal current of the motor is less than 1/6 of the nominal output current of the drive
- If the drive is used without a motor connected (for example, for test purposes)
- If the drive runs a medium-voltage motor through a step-up transformer.

In scalar control, some standard features are not available.

IR compensation for a scalar controlled drive

IR compensation is active only when the motor control mode is scalar. When IR compensation is activated, the drive gives an extra voltage boost to the motor at low speeds. IR compensation is useful in applications that require a high break-away torque.

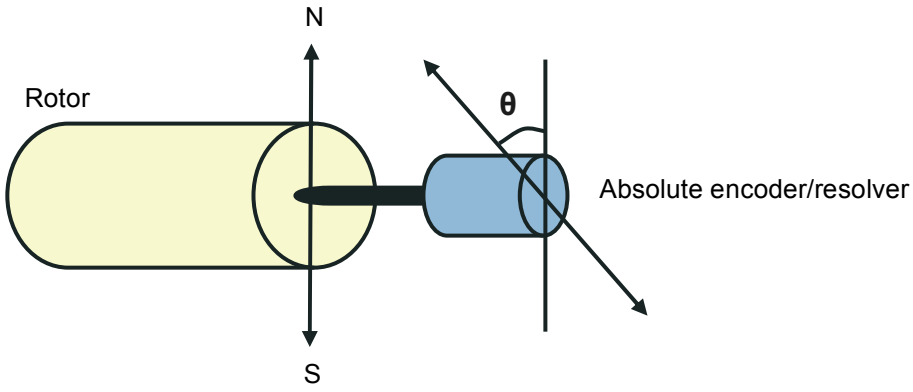
In direct torque control (DTC) mode, IR compensation is automatic and manual adjustment is not needed.



Autophasing

Autophasing is an automatic measurement routine to determine the angular position of the magnetic flux of a permanent magnet synchronous motor. The motor control requires the absolute position of the rotor flux in order to control motor torque accurately.

Sensors like absolute encoders and resolvers indicate the rotor position at all times after the offset between the zero angle of rotor and that of the sensor has been established. On the other hand, a standard pulse encoder determines the rotor position when it rotates but the initial position is not known. However, a pulse encoder can be used as an absolute encoder if it is equipped with Hall sensors, albeit with coarse initial position accuracy. The Hall sensors generate so-called commutation pulses that change their state six times during one revolution, so it is only known within which 60° sector of a complete revolution the initial position is.



The autophasing routine is performed with permanent magnet synchronous motors in the following cases:

1. One-time measurement of the rotor and encoder position difference when an absolute encoder, a resolver, or an encoder with commutation signals is used
2. At every power-up when an incremental encoder is used
3. With open-loop motor control, repetitive measurement of the rotor position at every start.

Several autophasing modes are available (see parameter 11.07 AUTOPHASING MODE).

The turning mode is recommended especially with case 1 as it is the most robust and accurate method. In turning mode, the motor shaft is turned back and forward (±360/polepairs)° in order to determine the rotor position. In case 2 (open-loop control), the shaft is turned only in one direction and the angle is smaller.

The standstill modes can be used if the motor cannot be turned (for example, when the load is connected). As the characteristics of motors and loads differ, testing must be done to find out the most suitable standstill mode.

The drive is also capable of determining the rotor position when started to a running motor in open-loop or closed-loop modes. In this situation, the setting of 11.07 AUTOPHASING MODE has no effect.

Thermal motor protection

With the parameters in group 45 MOT THERM PROT, the user can set up the motor overtemperature protection and configure motor temperature measurement (if present). This block also shows the estimated and measured motor temperature.

The motor can be protected against overheating by

- the motor thermal protection model
- measuring the motor temperature with PTC or KTY84 sensors. This will result in a more accurate motor model.

Thermal motor protection model

The drive calculates the temperature of the motor on the basis of the following assumptions:

1) When power is applied to the drive for the first time, the motor is at ambient temperature (defined by parameter 45.05 AMBIENT TEMP). After this, when power is applied to the drive, the motor is assumed to be at the estimated temperature (value of 1.18 MOTOR TEMP EST, saved at power switch-off).

2) Motor temperature is calculated using the user-adjustable motor thermal time and motor load curve. The load curve should be adjusted in case the ambient temperature exceeds 30 °C.

It is possible to adjust the motor temperature supervision limits and select how the drive reacts when overtemperature is detected.

Note: The motor thermal model can be used when only one motor is connected to the inverter.

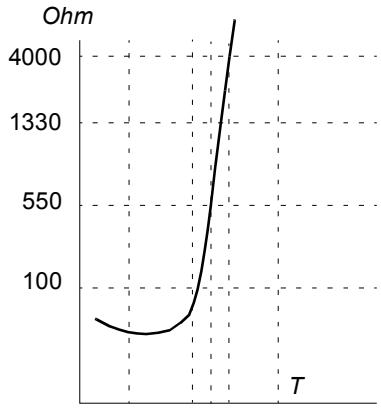
Temperature sensors

It is possible to detect motor overtemperature by connecting a motor temperature sensor to thermistor input TH of the drive or to optional encoder interface module FEN-xx.

Constant current is fed through the sensor. The resistance of the sensor increases as the motor temperature rises over the sensor reference temperature T_{ref}, as does the voltage over the resistor. The temperature measurement function reads the voltage and converts it into ohms.

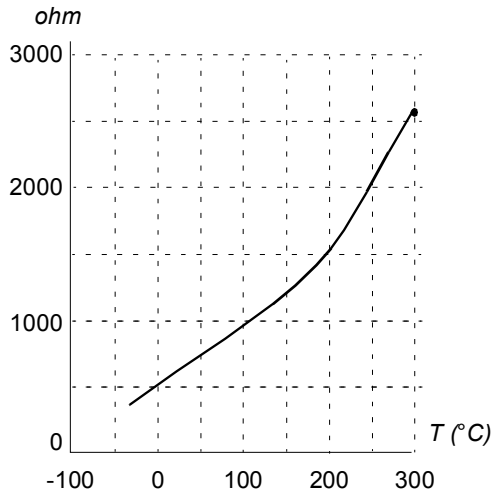
The figure below shows typical PTC sensor resistance values as a function of the motor operating temperature.

Temperature	PTC resistance
Normal	0…1.5 kohm
Excessive	≥ 4 kohm



The figure below shows typical KTY84 sensor resistance values as a function of the motor operating temperature.

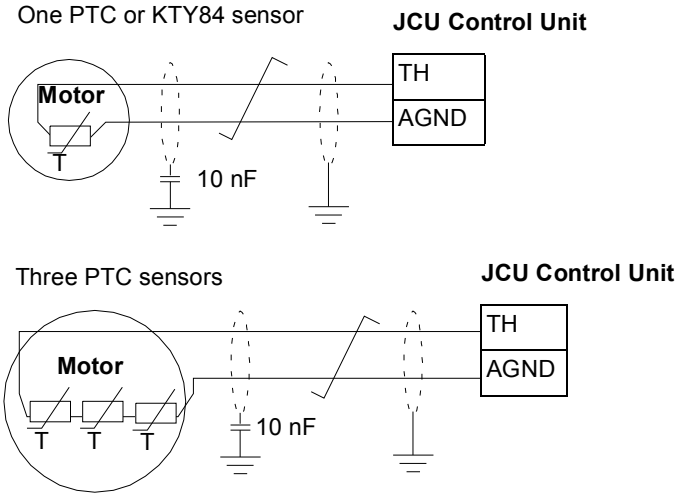
KTY84 scaling
90 °C = 936 ohm
110 °C = 1063 ohm
130 °C = 1197 ohm
150 °C = 1340 ohm



It is possible to adjust the motor temperature supervision limits and select how the drive reacts when overtemperature is detected.

WARNING! As the thermistor input on the JCU Control Unit is not insulated according to IEC 60664, the connection of the motor temperature sensor requires double or reinforced insulation between motor live parts and the sensor. If the assembly does not fulfil the requirement,
- the I/O board terminals must be protected against contact and must not be connected to other equipment
or
- the temperature sensor must be isolated from the I/O terminals.

The figure below shows a motor temperature measurement when thermistor input TH is used.



For encoder interface module FEN-xx connection, see the *User's Manual* of the appropriate encoder interface module.

DC voltage control features

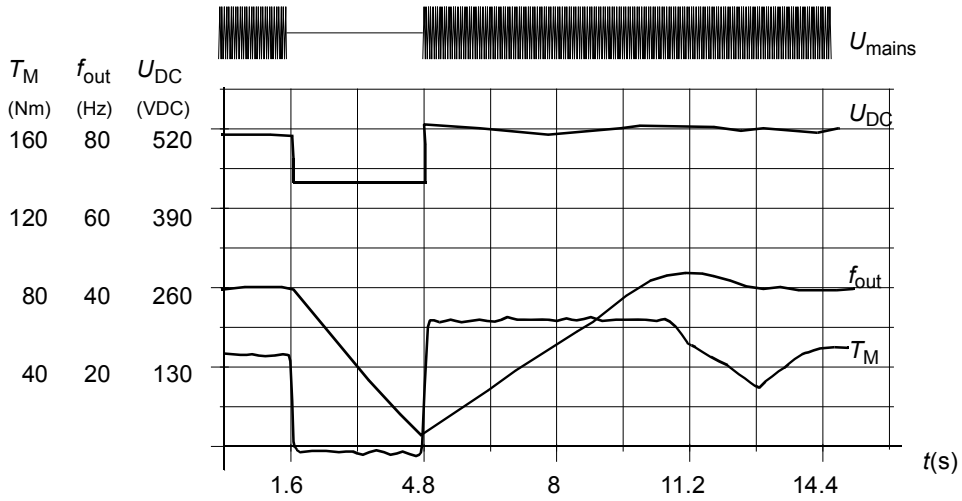
Overvoltage control

Overvoltage control of the intermediate DC link is needed with two-quadrant line-side converters when the motor operates within the generating quadrant. To prevent the DC voltage from exceeding the overvoltage control limit, the overvoltage controller automatically decreases the generating torque when the limit is reached.

Undervoltage control

If the incoming supply voltage is cut off, the drive will continue to operate by utilising the kinetic energy of the rotating motor. The drive will be fully operational as long as the motor rotates and generates energy to the drive. The drive can continue the operation after the break if the main contactor remained closed.

Note: Units equipped with main contactor option must be equipped with a hold circuit (e.g. UPS) which keeps the contactor control circuit closed during a short supply break.



U_{DC}= intermediate circuit voltage of the drive, f_{out} = output frequency of the drive, T_M = motor torque

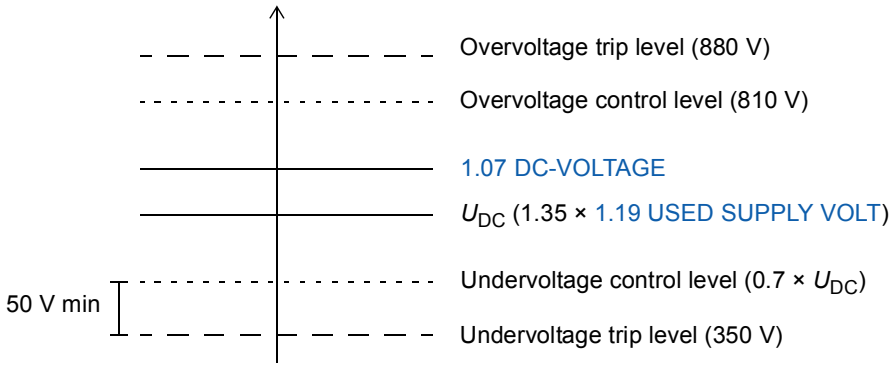
Loss of supply voltage at nominal load (f_{out} = 40 Hz). The intermediate circuit DC voltage drops to the minimum limit. The controller keeps the voltage steady as long as the supply is switched off. The drive runs the motor in generator mode. The motor speed falls but the drive is operational as long as the motor has enough kinetic energy.

Voltage control and trip limits

The control and trip limits of the intermediate DC voltage regulator are relative either to a supply voltage value provided by the user or to an automatically-determined supply voltage. The actual voltage used is shown by parameter 1.19 USED SUPPLY VOLT. The DC voltage (U_{DC}) equals 1.35 times this value.

Automatic identification of the supply voltage is performed every time the drive is powered. Automatic identification can be disabled by parameter 47.03

SUPPLVOLTAUTO-ID; the user can define the voltage manually at parameter 47.04 SUPPLY VOLTAGE.


Overvoltage trip level (880 V)
Overvoltage control level (810 V)
1.07 DC-VOLTAGE
U_{DC} (1.35 × 1.19 USED SUPPLY VOLT)
Undervoltage control level (0.7 × U_{DC})
50 V min
Undervoltage trip level (350 V)

The intermediate DC circuit is charged over an internal resistor which is bypassed when the correct level (80% of U_{DC}) is reached and voltage is stabilised.

Braking chopper

The built-in braking chopper of the drive can be used to handle the energy generated by a decelerating motor.

When the braking chopper is enabled and a resistor connected, the chopper will start conducting when the DC link voltage of the drive reaches 780 V. The maximum braking power is achieved at 840 V.

Low voltage mode

A Low voltage mode is available to extend the supply voltage range. When the mode is enabled, the drive can operate below the nominal range, for example when it needs to be powered from an emergency supply.

Low voltage mode can be activated by parameter 47.05 LOW VOLT MOD ENA. Low voltage mode introduces parameters 47.06 LOW VOLT DC MIN and 47.07 LOW VOLT DC MAX for adjustment of minimum and maximum DC voltage control levels respectively. The following rules apply:

- 47.06 LOW VOLT DC MIN = 250 to 450 V
- 47.07 LOW VOLT DC MAX = 350 to 810 V
- 47.07 LOW VOLT DC MAX > 47.06 LOW VOLT DC MIN + 50 V.

The value of parameter 47.08 EXT PU SUPPLY or its source should be set to 1 (true) when a supply below 270 V DC – such as a battery – is used. In such a configuration, an additional DC power supply (JPO-01) is needed to power the main circuit electronics. With an AC supply, the value of parameter 47.08 EXT PU SUPPLY or its source should be set to 0 (false).

Parameters 47.06…47.08 are effective only when Low voltage mode is active, i.e. value of parameter 47.05 LOW VOLT MOD ENA (or its source) is 1.

In Low voltage mode, the default voltage control and trip levels as well as the braking chopper operation levels (see sections *Voltage control and trip limits* and *Braking chopper* elsewhere in this chapter) are changed as follows:

Level	Value of parameter 47.08 EXT PU SUPPLY	
	FALSE	TRUE
Supply voltage range	200…240 V AC ±10% 270…324 V DC ±10%	*48…270 V DC ±10%
Overvoltage trip level	Unaffected	Unaffected
Overvoltage control level	47.07 LOW VOLT DC MAX	47.07 LOW VOLT DC MAX
Undervoltage control level	47.06 LOW VOLT DC MIN	Disabled
Undervoltage trip level	47.06 LOW VOLT DC MIN - 50 V	Disabled
Braking chopper activation level	47.07 LOW VOLT DC MAX - 30 V	47.07 LOW VOLT DC MAX - 30 V
Braking chopper maximum power level	47.07 LOW VOLT DC MAX + 30 V	47.07 LOW VOLT DC MAX + 30 V
*Requires additional DC power supply JPO-01		

Different system configurations are detailed in *ACSM1 System Engineering Manual* (3AFE68978297 [English]).

Speed control features

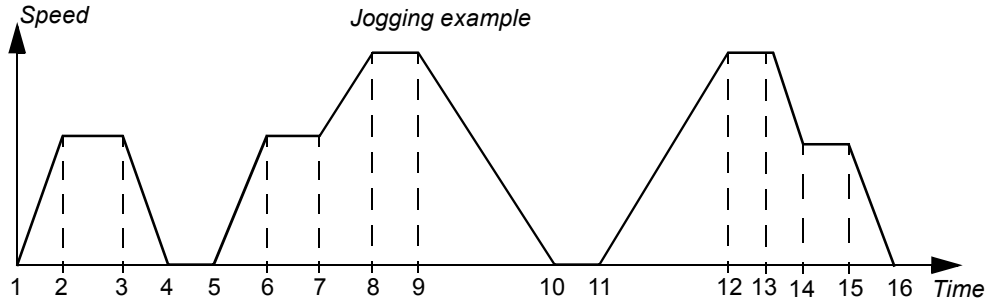
Jogging

Jogging is typically used during servicing or commissioning to control the machinery locally. It involves rotating the motor in small increments until the desired load position is achieved.

Two jogging functions (1 or 2) are available. When a jogging function is activated, the drive starts and accelerates to the defined jogging speed (parameters 24.10 SPEED REF JOG1 and 24.11 SPEED REF JOG2) along the defined jogging acceleration ramp. When the function is deactivated, the drive decelerates to a stop along the defined jogging deceleration ramp. One push button can be used to start and stop the drive during jogging.

Jogging functions 1 and 2 are activated by a parameter or through fieldbus. The source of the jogging command is selected by bit pointer parameters 10.07 JOG1 START and 10.14 JOG2 START. For activation through fieldbus, see 2.12 FBA MAIN CW.

The figure and table below describe the operation of the drive during jogging. (Note that they cannot be directly applied to jogging commands through fieldbus as those require no enable signal; see parameter 10.15 JOG ENABLE.) They also represent how the drive shifts to normal operation (= jogging inactive) when the drive start command is switched on. Jog cmd = State of the jogging input; Jog enable = Jogging enabled by the source set by parameter 10.15 JOG ENABLE; Start cmd = State of the drive start command.



Phase	Jog cmd	Jog enable	Start cmd	Description
1-2	1	1	0	Drive accelerates to the jogging speed along the acceleration ramp of the jogging function.
2-3	1	1	0	Drive runs at the jogging speed.
3-4	0	1	0	Drive decelerates to zero speed along the deceleration ramp of the jogging function.
4-5	0	1	0	Drive is stopped.
5-6	1	1	0	Drive accelerates to the jogging speed along the acceleration ramp of the jogging function.
6-7	1	1	0	Drive runs at the jogging speed.
7-8	x	0	1	Jog enable is not active; normal operation continues.
8-9	x	0	1	Normal operation overrides the jogging. Drive follows the speed reference.
9-10	x	0	0	Drive decelerates to zero speed along the active deceleration ramp.
10-11	x	0	0	Drive is stopped.

Phase	Jog cmd	Jog enable	Start cmd	Description
11-12	x	0	1	Normal operation overrides the jogging. Drive accelerates to the speed reference along the active acceleration ramp.
12-13	1	1	1	Start command overrides the jog enable signal.
13-14	1	1	0	Drive decelerates to the jogging speed along the deceleration ramp of the jogging function.
14-15	1	1	0	Drive runs at the jogging speed.
15-16	x	0	0	Drive decelerates to zero speed along the deceleration ramp of the jogging function.

Notes:

- Jogging is not operational when the drive start command is on, or when the drive is in local control.
- Normal start is inhibited when jog enable is active.
- The ramp shape time is set to zero during jogging.

Speed controller tuning

The speed controller of the drive can be automatically adjusted using the autotune function (parameter 28.16 PI TUNE MODE). Autotuning is based on the load and inertia of the motor and the machine. It is, however, also possible to manually adjust the controller gain, integration time and derivation time.

Automatic autotuning can be performed in four different ways depending on the setting of parameter 28.16 PI TUNE MODE. The selections (1) Smooth, (2) Middle and (3) Tight define how the drive torque reference should react to a speed reference step after tuning. The selection (1) Smooth will produce a slow response; (3) Tight will produce a fast response. The selection (4) User allows customised control sensibility adjustment through parameters 28.17 TUNE BANDWIDTH and 28.18 TUNE DAMPING. Detailed tuning status information is provided by parameter 6.03 SPEED CTRL STAT. If the autotuning routine fails, the SPEED CTRL TUNE FAIL alarm will occur for approximately 15 seconds. If a stop command is given to the drive during the autotuning, the routine is aborted.

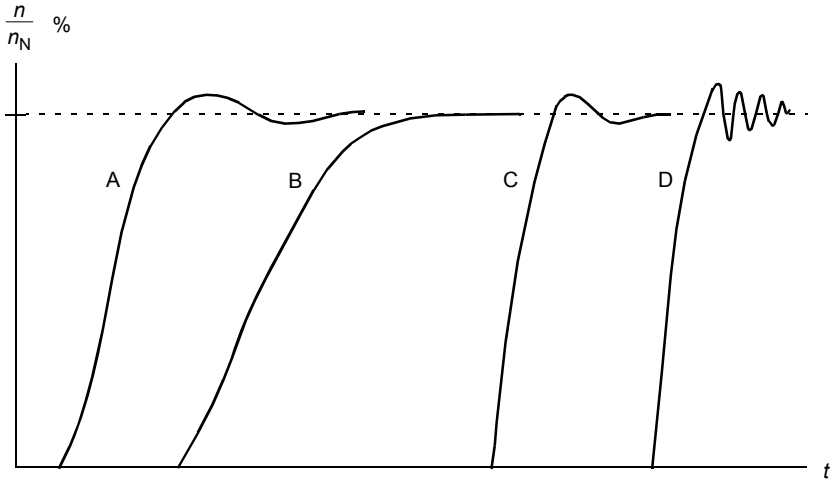
The prerequisites for performing the autotune routine are:

- The ID run has been successfully completed
- Speed, torque, current and acceleration limits (parameter groups 20 LIMITS and 25 SPEED REF RAMP) are set
- Speed feedback filtering, speed error filtering and zero speed are set (parameter groups 22 SPEED FEEDBACK and 26 SPEED ERROR)
- The drive is stopped.

The results of the autotune routine are automatically transferred into parameters

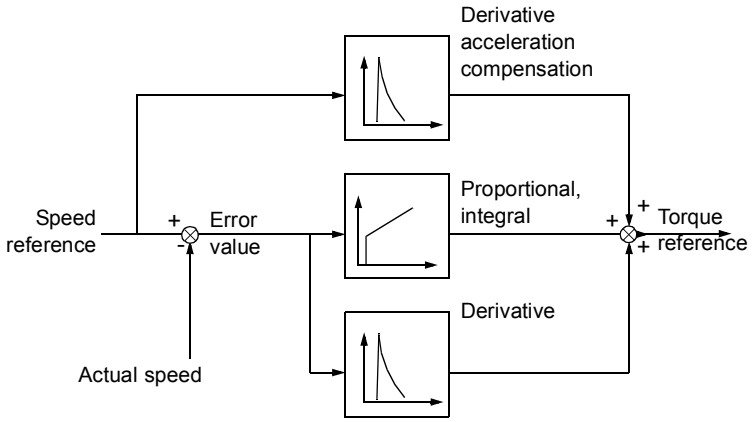
- 28.02 PROPORT GAIN (proportional gain of the speed controller)
- 28.03 INTEGRATION TIME (integration time of the speed controller)
- 1.31 MECH TIME CONST (mechanical time constant of the machinery).

The figure below shows speed responses at a speed reference step (typically 1…20%).



A: Undercompensated
B: Normally tuned (autotuning)
C: Normally tuned (manually). Better dynamic performance than with B
D: Overcompensated speed controller

The figure below is a simplified block diagram of the speed controller. The controller output is the reference for the torque controller.



For more information on the use of the autotune function, see the description of parameter 28.16 PI TUNE MODE.

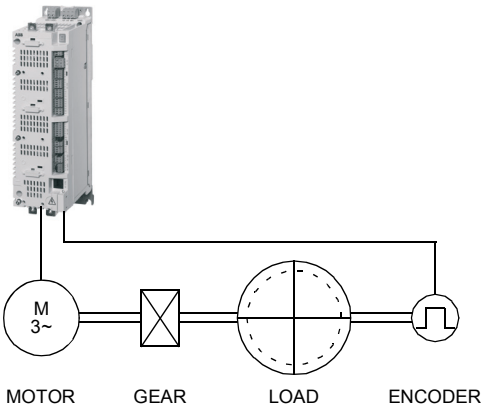
Motor feedback features

Motor encoder gear function

The drive provides motor encoder gear function for compensating of mechanical gears between the motor shaft, the encoder and the load.

Motor encoder gear application example:

Speed control uses the motor speed. If no encoder is mounted on the motor shaft, the motor encoder gear function must be applied in order to calculate the actual motor speed on the basis of the measured load speed.



The motor encoder gear parameters 22.03 MOTOR GEAR MUL and 22.04 MOTOR GEAR DIV are set as follows:

$$\frac{\text{22.03 MOTOR GEAR MUL}}{\text{22.04 MOTOR GEAR DIV}} = \frac{\text{Actual speed}}{\text{Encoder 1/2 speed}}$$

Note: If the motor gear ratio differs from 1, the motor model uses an estimated speed instead of the speed feedback value.

See also section *Examples of gear function usage* on page 58.

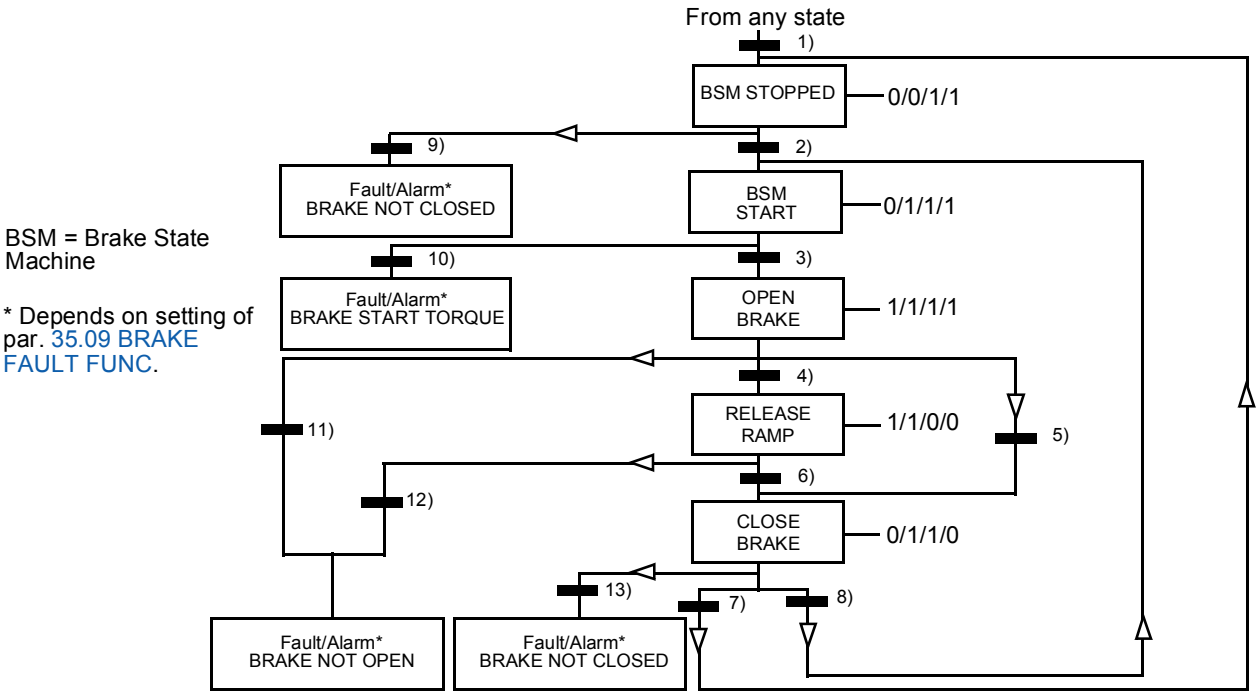
Mechanical brake control

The program supports the use of a mechanical brake to hold the motor and load at zero speed when the drive is stopped or not powered.

Mechanical brake control (with or without acknowledgement) is activated by parameter 35.01 BRAKE CONTROL. The acknowledgement (supervision) signal can be connected to, for example, a digital input. The brake on/off value is reflected by 3.15 BRAKE COMMAND, which should be connected to a relay (or digital) output. The brake will open upon drive start after the delay 35.03 BRAKE OPEN DELAY has elapsed and requested motor start torque 35.06 BRAKE OPEN TORQ is available. The brake will close after motor speed decreases below 35.05 BRAKE CLOSE SPD and the delay 35.04 BRAKE CLOSE DLY has elapsed. When the brake close command is issued, the motor torque is stored into 3.14 BRAKE TORQ MEM.

Note: The mechanical brake must be opened manually before the ID run.

Mechanical brake state diagram



State (Symbol NN W/X/Y/Z)

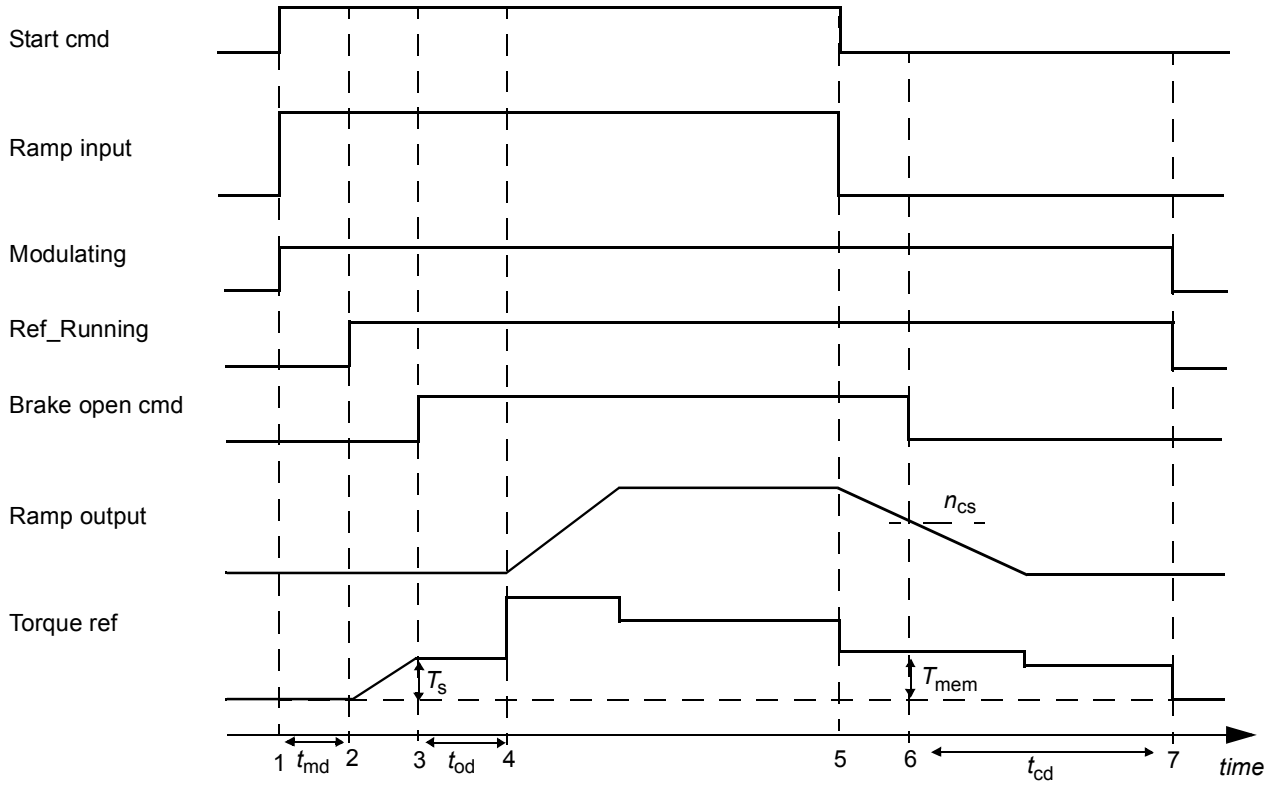
- NN: State name
- W/X/Y/Z: State outputs/operations
  - W: 1 = Brake open command is active. 0 = Brake close command is active. (Controlled through selected digital/relay output with signal 3.15 BRAKE COMMAND.)
  - X: 1 = Forced start (inverter is modulating). The function keeps the internal Start command on until the brake is closed regardless of the status of the external Stop. Effective only when ramp stop has been selected as the stop mode (11.03 STOP MODE). Run enable and faults override the forced start. 0 = No forced start (normal operation).
  - Y: 1 = Drive control mode is forced to speed/scalar.
  - Z: 1 = Ramp generator output is forced to zero. 0 = Ramp generator output is enabled (normal operation).

State change conditions (Symbol ▬)

1) Brake control is active (35.01 BRAKE CONTROL = (1) WITH ACK or (2) NO ACK) OR modulation of the drive is requested to stop. The drive control mode is forced to speed/scalar.
2) External start command is on AND brake open request is on (source selected by 35.07 BRAKE CLOSE REQ = 0)..
3) Starting torque required at brake release is reached (35.06 BRAKE OPEN TORQ) AND brake hold is not active (35.08 BRAKE OPEN HOLD). **Note:** With scalar control, the defined starting torque has no effect.
4) Brake is open (acknowledgement = 1, selected by par. 35.02 BRAKE ACKNOWL) AND the brake open delay has passed (35.03 BRAKE OPEN DELAY). Start = 1.
5) 6) Start = 0 OR brake close command is active AND actual motor speed < brake close speed (35.05 BRAKE CLOSE SPD).
7) Brake is closed (acknowledgement = 0) AND brake close delay has passed (35.04 BRAKE CLOSE DLY). Start = 0.
8) Start = 1.
9) Brake is open (acknowledgement = 1) AND brake close delay has passed.
10) Defined starting torque at brake release is not reached.
11) Brake is closed (acknowledgement = 0) AND brake open delay has passed.
12) Brake is closed (acknowledgement = 0).
13) Brake is open (acknowledgement = 1) AND brake close delay has passed.

Operation time scheme

The simplified time scheme below illustrates the operation of the brake control function.



T_s	Start torque at brake release (parameter 35.06 BRAKE OPEN TORQ)
T_{mem}	Stored torque value at brake close (signal 3.14 BRAKE TORQ MEM)
t_{md}	Motor magnetising delay
t_{od}	Brake open delay (parameter 35.03 BRAKE OPEN DELAY)
n_{cs}	Brake close speed (parameter 35.05 BRAKE CLOSE SPD)
t_{cd}	Brake close delay (parameter 35.04 BRAKE CLOSE DLY)

Example

The figure below shows a brake control application example.

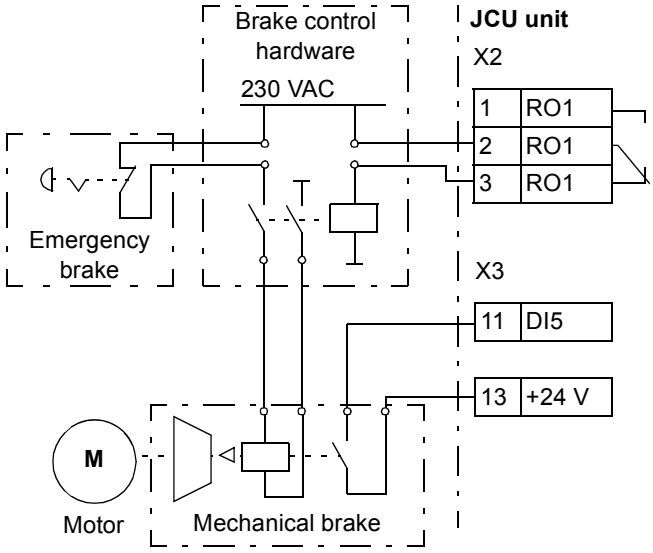
WARNING! Make sure that the machinery into which the drive with brake control function is integrated fulfils the personnel safety regulations. Note that the frequency converter (a Complete Drive Module or a Basic Drive Module, as defined in IEC 61800-2), is not considered as a safety device mentioned in the European Machinery Directive and related harmonised standards. Thus, the personnel safety of the complete machinery must not be based on a specific frequency converter feature (such as the brake control function), but it has to be implemented as defined in the application specific regulations.

The brake on/off is controlled via signal 3.15 BRAKE COMMAND. The source for the brake supervision is selected by parameter 35.02 BRAKE ACKNOWL.

The brake control hardware and wirings need to be done by the user.

- Brake on/off control through selected relay/digital output.
- Brake supervision through selected digital input.
- Emergency brake switch in the brake control circuit.

- Brake on/off control through relay output (i.e. parameter 12.12 RO1 OUT PTR is set to P.03.15 = 3.15 BRAKE COMMAND).
- Brake supervision through digital input DI5 (i.e. parameter 35.02 BRAKE ACKNOWL is set to P.02.01.04 = 2.01 DI STATUS bit 4)



Emergency stop

Note: The user is responsible for installing the emergency stop devices and all the additional devices needed for the emergency stop to fulfil the required emergency stop category classes.

The emergency stop signal is connected to the digital input which is selected as the source for the emergency stop activation (parameter 10.10 EM STOP OFF3 or 10.11 EM STOP OFF1). Emergency stop can also be activated through fieldbus (2.12 FBA MAIN CW).

Note: When an emergency stop signal is detected, the emergency stop function cannot be cancelled even though the signal is cancelled.

For more information, refer to *Application Guide: Functional Safety Solutions with ACSM1 Drives* (3AUA0000031517 [English]).

Miscellaneous features

Backup and restore of drive contents

General

The drive offers a possibility of backing up numerous settings and configurations to external storage such as a PC file (using the DriveStudio tool) and the internal memory of the control panel. These settings and configurations can then be restored to the drive, or a number of drives.

Backup using DriveStudio includes

- Parameter settings
- User parameter sets
- Application program.

Backup using the drive control panel includes

- Parameter settings
- User parameter sets.

For detailed instructions for performing the backup/restore, refer to the DriveStudio and control panel documentation.

Limitations

A backup can be done without interfering with drive operation, but restoring a backup always resets and reboots the control unit, so restore is not possible with the drive running.

Backup/restore between different program variants (such as the Motion Control Program and the Speed and Torque Control Program) is not possible.

Restoring backup files from one firmware version to another is considered risky, so the results should be carefully observed and verified when done for the first time. The parameters and application support are bound to change between firmware versions and backups are not always compatible with other firmware versions even if

restore is allowed by the backup/restore tool. Before using the backup/restore functions between different firmware versions, refer to the release notes of each version.

Applications should not be transferred between different firmware versions. Contact the supplier of the application when it needs to be updated for a new firmware version.

Parameter restore

Parameters are divided into three different groups that can be restored together or individually:

- Motor configuration parameters and identification (ID) run results
- Fieldbus adapter and encoder settings
- Other parameters.

For example, retaining the existing ID run results in the drive will make a new ID run unnecessary.

Restore of individual parameters can fail for the following reasons:

- The restored value does not fall within the minimum and maximum limits of the drive parameter
- The type of the restored parameter is different from that in the drive
- The restored parameter does not exist in the drive (often the case when restoring the parameters of a new firmware version to a drive with an older version)
- The backup does not contain a value for the drive parameter (often the case when restoring the parameters of an old firmware version to a drive with a newer version).

In these cases, the parameter is not restored; the backup/restore tool will warn the user and offers a possibility to set the parameter manually.

User parameter sets

The drive has four user parameter sets that can be saved to the permanent memory and recalled using drive parameters. It is also possible to use digital inputs to switch between different user parameter sets. See the descriptions of parameters 16.09…16.12.

A user parameter set contains all values of parameter groups 10 to 99 (except the fieldbus communication configuration settings).

As the motor settings are included in the user parameter sets, make sure the settings correspond to the motor used in the application before recalling a user set. In an application where different motors are used with one drive, the motor ID run needs to be performed with each motor and saved to different user sets. The appropriate set can then be recalled when the motor is switched.

Default connections of the control unit

What this chapter contains

This chapter shows the default control connections of the JCU Control Unit.

More information on the connectivity of the JCU is given in the *Hardware Manual* of the drive.

Notes:

*Total maximum current: 200 mA

1) Selected by par. 12.01 DIO1 CONF.

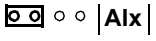
2) Selected by par. 12.02 DIO2 CONF.

3) Selected by par. 12.03 DIO3 CONF.

4) Selected by jumper J1.

5) Selected by jumper J2.

Current:

J1/2 

Voltage:

J1/2

		X1
External power input 24 V DC, 1.6 A	+24VI	1
	GND	2
		X2
Relay output: Brake close/open 250 V AC / 30 V DC 2 A	NO	1
	COM	2
	NC	3
		X3
+24 V DC*	+24VD	1
Digital I/O ground	DGND	2
Digital input 1: Stop/start (par. 10.02 and 10.05)	DI1	3
Digital input 2: EXT1/EXT2 (par. 34.01)	DI2	4
+24 V DC*	+24VD	5
Digital I/O ground	DGND	6
Digital input 3: Fault reset (par. 10.08)	DI3	7
Digital input 4: Not connected	DI4	8
+24 V DC*	+24VD	9
Digital I/O ground	DGND	10
Digital input 5: Not connected	DI5	11
Digital input 6: Not connected	DI6	12
+24 V DC*	+24VD	13
Digital I/O ground	DGND	14
Digital input/output 1 [1]: Ready	DIO1	15
Digital input/output 2 [2]: Running	DIO2	16
+24 V DC*	+24VD	17
Digital I/O ground	DGND	18
Digital input/output 3 [3]: Fault	DIO3	19
		X4
Reference voltage (+)	+VREF	1
Reference voltage (–)	–VREF	2
Ground	AGND	3
Analogue input 1 (mA or V) [4]: Speed reference (par. 24.01)	AI1+	4
	AI1–	5
Analogue input 2 (mA or V) [5]: Torque reference (par. 32.01)	AI2+	6
	AI2–	7
AI1 current/voltage selection		J1
AI2 current/voltage selection		J2
Thermistor input	TH	8
Ground	AGND	9
Analogue output 1 (mA): Output current	AO1 (I)	10
Analogue output 2 (V): Actual speed	AO2 (U)	11
Ground	AGND	12
		X5
Drive-to-drive link termination		J3
Drive-to-drive link	B	1
	A	2
	BGND	3
		X6
Safe Torque Off. Both circuits must be closed for the drive to start. See the appropriate drive hardware manual.	OUT1	1
	OUT2	2
	IN1	3
	IN2	4
Control panel connection		**X7**

Parameters and firmware blocks

What this chapter contains

This chapter lists and describes the parameters provided by the firmware.

Types of parameters

Parameters are user-adjustable operation instructions of the drive (groups 10…99). There are four basic types of parameters: Actual signals, value parameters, value pointer parameters and bit pointer parameters.

Actual signal

Type of parameter that is the result of a measurement or calculation by the drive. Actual signals can be monitored, but not adjusted, by the user. Actual signals are typically contained within parameter groups 1…9.

For additional actual signal data, e.g. update cycles and fieldbus equivalents, see chapter *Parameter data*.

Value parameter

A value parameter has a fixed set of choices or a setting range.

Example 1: Motor phase loss supervision is activated by selecting (1) Fault from the selection list of parameter 46.04 MOT PHASE LOSS.

Example 2: The motor nominal power (kW) is set by writing the appropriate value to parameter 99.10 MOT NOM POWER, e.g. 10.

Value pointer parameter

A value pointer parameter points to the value of another parameter. The source parameter is given in format **P.xx.yy**, where xx = Parameter group; yy = Parameter index. In addition, value pointer parameters may have a set of pre-selected choices.

Example: Motor current signal, 1.05 CURRENT PERC, is connected to analogue output AO1 by setting parameter 15.01 AO1 PTR to value P.01.05.

Bit pointer parameter

A bit pointer parameter points to the value of a bit in another parameter, or can be fixed to 0 (FALSE) or 1 (TRUE). In addition, bit pointer parameters may have a set of pre-selected choices.

When adjusting a bit pointer parameter on the optional control panel, CONST is selected in order to fix the value to 0 (displayed as “C.FALSE”) or 1 (“C.TRUE”). POINTER is selected to define a source from another parameter.

A pointer value is given in format **P.xx.yy.zz**, where xx = Parameter group, yy = Parameter index, zz = Bit number.

Example: Digital input DI5 status, 2.01 DI STATUS bit 4, is used for brake supervision by setting parameter 35.02 BRAKE ACKNOWL to value P.02.01.04.

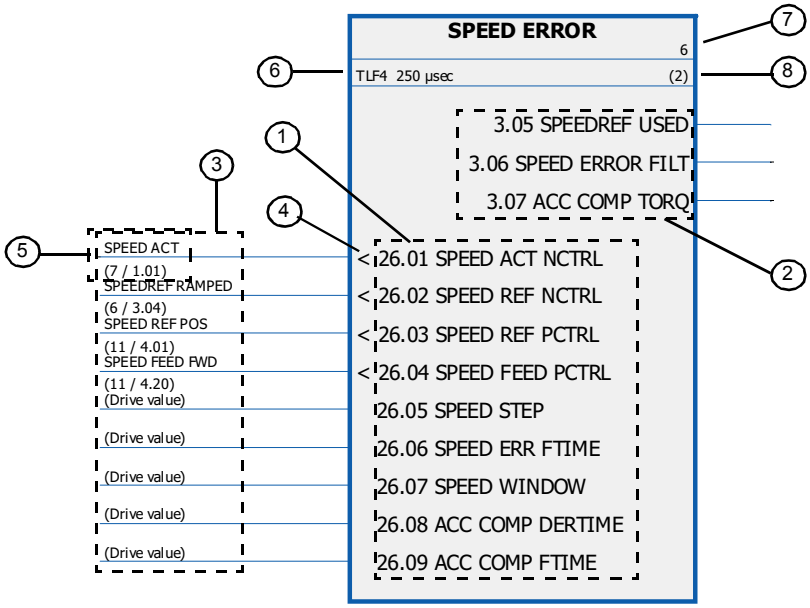
Note: Pointing to a nonexisting bit will be interpreted as 0 (FALSE).

For additional parameter data, e.g. update cycles and fieldbus equivalents, see chapter *Parameter data*.

Firmware blocks

Firmware blocks accessible from the DriveSPC PC tool are described in the parameter group that contains the most of the block inputs/outputs. Whenever a block has inputs or outputs outside the current parameter group, a reference is given. Likewise, parameters have a reference to the firmware block they are included in (if any).

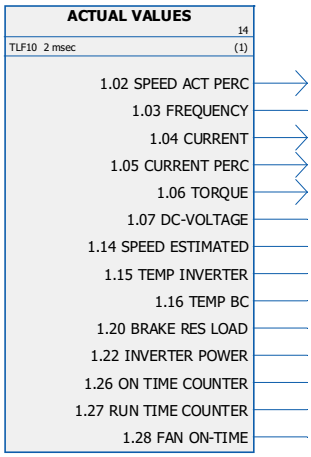
Note: Not all parameters are available through firmware blocks.



1	Inputs
2	Outputs
3	Input parameter values
4	Pointer parameter indicator “<“
5	Parameter 26.01 is set to value P.1.1, i.e. parameter 1.01 SPEED ACT. The “7” means the parameter can be found on page 7 of DriveSPC.
6	Information of the block internal execution order (“TLF4”) and time level (“250 µsec”). Time level, i.e. update cycle, is application-specific. See the time level of the block in DriveSPC.
7	Firmware block ID number in the application program
8	Firmware block execution order for the selected update cycle ID

Group 01 ACTUAL VALUES

This group contains basic actual signals for monitoring the drive.

Firmware block: **ACTUAL VALUES** (1)		
1.01	SPEED ACT	FW block: SPEED FEEDBACK (page 105)
	Filtered actual speed in rpm. Used speed feedback is defined by parameter 22.01 SPEED FB SEL. Filter time constant can be adjusted by parameter 22.02 SPEED ACT FTIME.	
1.02	SPEED ACT PERC	FW block: ACTUAL VALUES (see above)
	Actual speed in percent of the motor synchronous speed.	
1.03	FREQUENCY	FW block: ACTUAL VALUES (see above)
	Estimated drive output frequency in Hz.	
1.04	CURRENT	FW block: ACTUAL VALUES (see above)
	Measured motor current in A.	
1.05	CURRENT PERC	FW block: ACTUAL VALUES (see above)
	Motor current in percent of the nominal motor current.	
1.06	TORQUE	FW block: ACTUAL VALUES (see above)
	Motor torque in percent of the motor nominal torque.	
1.07	DC-VOLTAGE	FW block: ACTUAL VALUES (see above)
	Measured intermediate circuit voltage in V.	
1.08	ENCODER 1 SPEED	FW block: ENCODER (page 169)
	Encoder 1 speed in rpm.	
1.09	ENCODER 1 POS	FW block: ENCODER (page 169)
	Actual position of encoder 1 within one revolution.	

1.10	ENCODER 2 SPEED	FW block: ENCODER (page 169)
	Encoder 2 speed in rpm.	
1.11	ENCODER 2 POS	FW block: ENCODER (page 169)
	Actual position of encoder 2 within one revolution.	
1.14	SPEED ESTIMATED	FW block: ACTUAL VALUES (see above)
	Estimated motor speed in rpm.	
1.15	TEMP INVERTER	FW block: ACTUAL VALUES (see above)
	Measured temperature of the heatsink in Celsius.	
1.16	TEMP BC	FW block: ACTUAL VALUES (see above)
	Brake chopper IGBT temperature in Celsius.	
1.17	MOTOR TEMP	FW block: MOT THERM PROT (page 145)
	Measured motor temperature in Celsius when a KTY sensor is used. (With a PTC sensor, the value is always 0.)	
1.18	MOTOR TEMP EST	FW block: MOT THERM PROT (page 145)
	Estimated motor temperature in Celsius.	
1.19	USED SUPPLY VOLT	FW block: VOLTAGE CTRL (page 152)
	Either the user-given supply voltage (parameter 47.04 SUPPLY VOLTAGE), or, if auto-identification is enabled by parameter 47.03 SUPPLVOLTAUTO-ID, the automatically determined supply voltage.	
1.20	BRAKE RES LOAD	FW block: ACTUAL VALUES (see above)
	Estimated temperature of the braking resistor. The value is given in percent of the temperature the resistor reaches when loaded with the power defined by parameter 48.04 BR POWER MAX CNT.	
1.21	CPU USAGE	FW block: None
	Microprocessor load in percent.	
1.22	INVERTER POWER	FW block: ACTUAL VALUES (see above)
	Drive output power in kilowatts.	
1.26	ON TIME COUNTER	FW block: ACTUAL VALUES (see above)
	This counter runs when the drive is powered. The counter can be reset using the DriveStudio tool.	
1.27	RUN TIME COUNTER	FW block: ACTUAL VALUES (see above)
	Motor run time counter. The counter run when the drive modulates. The counter can be reset using the DriveStudio tool.	
1.28	FAN ON-TIME	FW block: ACTUAL VALUES (see above)
	Running time of the drive cooling fan. Can be reset by entering 0.	

1.31	MECH TIME CONST	FW block: None
	Mechanical time constant of the drive and the machinery as determined by the speed controller autotune function. See parameter 28.16 PI TUNE MODE on page 127.	

Group 02 I/O VALUES

This group contains information on the I/Os of the drive.

2.01	DI STATUS	FW block: DI (page 89)
	Status word of the digital inputs. Example: 000001 = DI1 is on, DI2 to DI6 are off.	
2.02	RO STATUS	FW block: RO (page 89)
	Status of relay output. 1 = RO is energized.	
2.03	DIO STATUS	FW blocks: DIO1 (page 87), DIO2 (page 87), DIO3 (page 87)
	Status word of digital inputs/outputs DIO1…3. Example: 001 = DIO1 is on, DIO2 and DIO3 are off.	
2.04	AI1	FW block: AI1 (page 91)
	Analogue input AI1 value in V or mA. The type is selected with jumper J1 on the JCU Control Unit.	
2.05	AI1 SCALED	FW block: AI1 (page 91)
	Scaled value of analogue input AI1. See parameters 13.04 AI1 MAX SCALE and 13.05 AI1 MIN SCALE.	
2.06	AI2	FW block: AI2 (page 92)
	Analogue input AI2 value in V or mA. The type is selected with jumper J2 on the JCU Control Unit.	
2.07	AI2 SCALED	FW block: AI2 (page 92)
	Scaled value of analogue input AI2. See parameters 13.09 AI2 MAX SCALE and 13.10 AI2 MIN SCALE.	
2.08	AO1	FW block: AO1 (page 95)
	Analogue output AO1 value in mA	
2.09	AO2	FW block: AO2 (page 96)
	Analogue output AO2 value in V	
2.10	DIO2 FREQ IN	FW block: DIO2 (page 87)
	Frequency input value in Hz when DIO2 is used as frequency input (12.02 DIO2 CONF is set to (2) Freq input).	
2.11	DIO3 FREQ OUT	FW block: DIO3 (page 87)
	Frequency output value in Hz when DIO3 is used as frequency output (12.03 DIO3 CONF is set to (2) Freq output).	

2.12	FBA MAIN CW	FW block: FIELDBUS (page 156)

Control Word for fieldbus communication.

Log. = Logical combination (i.e. Bit AND/OR Selection parameter). Par. = Selection parameter. See *State diagram* on page *330.*

Bit	Name	Val.	Information	Log.	Par.
0	STOP*	1	Stop according to the stop mode selected by 11.03 STOP MODE or according to the requested stop mode (bits 2…6). **Note:** Simultaneous STOP and START commands result in a stop command.	OR	10.02, 10.03, 10.05, 10.06
		0	No operation		
1	START	1	Start. **Note:** Simultaneous STOP and START commands result in a stop command.	OR	10.02, 10.03, 10.05, 10.06
		0	No operation		
2	STPMODE EM OFF*	1	Emergency OFF2 (bit 0 must be 1): Drive is stopped by cutting off the motor power supply (the inverter IGBTs are blocked). The motor coasts to stop. The drive will restart only with the next rising edge of the start signal when the run enable signal is on.	AND	-
		0	No operation		
3	STPMODE EM STOP*	1	Emergency stop OFF3 (bit 0 must be 1): Stop within the time defined by 25.11 EM STOP TIME.	AND	10.10
		0	No operation		
4	STPMODE OFF1*	1	Emergency stop OFF1 (bit 0 must be 1): Stop along the currently active deceleration ramp.	AND	10.11
		0	No operation		
5	STPMODE RAMP*	1	Stop along the currently active deceleration ramp.	-	11.03
		0	No operation		
6	STPMODE COAST*	1	Coast to stop.	-	11.03
		0	No operation		
7	RUN ENABLE	1	Activate run enable.	AND	10.09
		0	Activate run disable.		
8	RESET	0->1	Fault reset if an active fault exists.	OR	10.08
		other	No operation		
9	JOGGING 1	1	Activate jogging function 1. See section *Jogging* on page *45*.	OR	10.07
		0	Jogging function 1 disabled		

* If all stop mode bits 2…6 are 0, stop mode is selected by 11.03 STOP MODE. Coast stop (bit 6) overrides the emergency stop (bit 2/3/4). Emergency stop overrides the normal ramp stop (bit 5).

2.12 FBA MAIN CW (continued from previous page)

Bit	Name	Val.	Information	Log.	Par.
10	JOGGING 2	1	Activate jogging function 2. See section *Jogging* on page *45*.	OR	10.14
		0	Jogging function 2 disabled		
11	REMOTE CMD	1	Fieldbus control enabled	-	-
		0	Fieldbus control disabled		
12	RAMP OUT 0	1	Force Ramp Function Generator output to zero. Drive ramps to a stop (current and DC voltage limits in force).	-	-
		0	No operation		
13	RAMP HOLD	1	Halt ramping (Ramp Function Generator output held).	-	-
		0	No operation		
14	RAMP IN 0	1	Force Ramp Function Generator input to zero.	-	-
		0	No operation		
15	EXT1 / EXT2	1	Switch to external control location EXT2.	OR	34.01
		0	Switch to external control location EXT1.		
16	REQ STARTINH	1	Activate start inhibit.	-	-
		0	No start inhibit		
17	LOCAL CTL	1	Request local control for Control Word. Used when the drive is controlled via PC tool or panel or through local fieldbus. - Local fieldbus: Transfer to fieldbus local control (control via fieldbus control word or reference). Fieldbus steals the control. - Panel or PC tool: Transfer to local control.	-	-
		0	Request external control.		
18	FBLOCAL REF	1	Request fieldbus local control.	-	-
		0	No fieldbus local control		
19…27	Not in use				
28	CW B28		Freely programmable control bits.	-	-
29	CW B29				
30	CW B30				
31	CW B31				

2.13 FBA MAIN SW — FW block: FIELDBUS (page 156)

Status Word for fieldbus communication. See *State diagram* on page *330.*

Bit	Name	Value	Information
0	READY	1	Drive is ready to receive start command.
		0	Drive is not ready.
1	ENABLED	1	External run enable signal is received.
		0	No external run enable signal is received.
2	RUNNING	1	Drive is modulating.
		0	Drive is not modulating.
3	REF RUNNING	1	Normal operation is enabled. Drive is running and following given reference.
		0	Normal operation is disabled. Drive is not following given reference (for example, modulating during magnetization).
4	EM OFF (OFF2)	1	Emergency OFF2 is active.
		0	Emergency OFF2 is inactive.
5	EM STOP (OFF3)	1	Emergency stop OFF3 (ramp stop) is active.
		0	Emergency OFF3 is inactive.
6	ACK STARTINH	1	Start inhibit is active.
		0	Start inhibit is inactive.
7	ALARM	1	An alarm is active. See chapter *Fault tracing*.
		0	No alarm is active.
8	AT SETPOINT	1	Drive is at setpoint. Actual value equals reference value (i.e. the difference between the actual speed and the speed reference is within the speed window defined by 26.07 SPEED WINDOW).
		0	Drive has not reached setpoint.
9	LIMIT	1	Operation is limited by any torque or current limit.
		0	Operation is within torque/current limits.
10	ABOVE LIMIT	1	Actual speed exceeds the defined limit, 22.07 ABOVE SPEED LIM.
		0	Actual speed is within the defined limits.
11	EXT2 ACT	1	External control location EXT2 is active.
		0	External control location EXT1 is active.
12	LOCAL FB	1	Fieldbus local control is active.
		0	Fieldbus local control is inactive.
13	ZERO SPEED	1	Drive speed is below limit set by par. 22.05 ZERO SPEED LIMIT.
		0	Drive has not reached zero speed limit.
14	REV ACT	1	Drive is running in reverse direction.
		0	Drive is running in forward direction.
15	Not in use		
16	FAULT	1	Fault is active. See chapter *Fault tracing*.
		0	No fault is active.
17	LOCAL PANEL	1	Local control is active, i.e. drive is controlled from PC tool or control panel.
		0	Local control is inactive.

<table>
<tr><td>2.13</td><td>FBA MAIN SW (continued from previous page)</td></tr>
<tr><td></td><td>
<table>
<tr><th>Bit</th><th>Name</th><th>Value</th><th>Information</th></tr>
<tr><td>18…26</td><td colspan="3">Not in use with Speed and Torque Control Program</td></tr>
<tr><td rowspan="2">27</td><td rowspan="2">REQUEST CTL</td><td>1</td><td>Control word is requested from fieldbus.</td></tr>
<tr><td>0</td><td>Control word is not requested from fieldbus.</td></tr>
<tr><td>28</td><td>SW B28</td><td></td><td rowspan="4">Programmable status bits (unless fixed by the used profile). See parameters 50.08…50.11 and the user manual of the fieldbus adapter.</td></tr>
<tr><td>29</td><td>SW B29</td><td></td></tr>
<tr><td>30</td><td>SW B30</td><td></td></tr>
<tr><td>31</td><td>SW B31</td><td></td></tr>
</table>
</td></tr>
<tr><td>2.14</td><td>FBA MAIN REF1</td><td>FW block: FIELDBUS (page 156)</td></tr>
<tr><td></td><td colspan="2">Scaled fieldbus reference 1. See parameter 50.04 FBA REF1 MODESEL.</td></tr>
<tr><td>2.15</td><td>FBA MAIN REF2</td><td>FW block: FIELDBUS (page 156)</td></tr>
<tr><td></td><td colspan="2">Scaled fieldbus reference 2. See parameter 50.05 FBA REF2 MODESEL.</td></tr>
<tr><td>2.16</td><td>FEN DI STATUS</td><td>FW block: ENCODER (page 169)</td></tr>
<tr><td></td><td colspan="2">Status of digital inputs of FEN-xx encoder interfaces in drive option Slots 1 and 2. Examples:
000001 (01h) = DI1 of FEN-xx in Slot 1 is ON, all others are OFF.
000010 (02h) = DI2 of FEN-xx in Slot 1 is ON, all others are OFF.
010000 (10h) = DI1 of FEN-xx in Slot 2 is ON, all others are OFF.
100000 (20h) = DI2 of FEN-xx in Slot 2 is ON, all others are OFF.</td></tr>
<tr><td>2.17</td><td>D2D MAIN CW</td><td>FW block: D2D COMMUNICATION (page 164)</td></tr>
<tr><td></td><td colspan="2">Drive-to-drive control word received through the drive-to-drive link. See also actual signal 2.18 below.
<table>
<tr><th>Bit</th><th>Information</th></tr>
<tr><td>0</td><td>Stop.</td></tr>
<tr><td>1</td><td>Start.</td></tr>
<tr><td>2…6</td><td>Reserved.</td></tr>
<tr><td>7</td><td>Run enable. By default, not connected in a follower drive.</td></tr>
<tr><td>8</td><td>Reset. By default, not connected in a follower drive.</td></tr>
<tr><td>9…14</td><td>Freely assignable through bit pointer parameters.</td></tr>
<tr><td>15</td><td>EXT1/EXT2 selection. 0 = EXT1 active, 1 = EXT2 active. By default, not connected in a follower drive.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>2.18</td><td>D2D FOLLOWER CW</td><td>FW block: DRIVE LOGIC (page 78)</td></tr>
<tr><td></td><td colspan="2">Drive-to-drive control word sent to the followers by default. See also firmware block D2D COMMUNICATION on page 164.
<table>
<tr><th>Bit</th><th>Information</th></tr>
<tr><td>0</td><td>Stop.</td></tr>
<tr><td>1</td><td>Start.</td></tr>
<tr><td>2…6</td><td>Reserved.</td></tr>
<tr><td>7</td><td>Run enable.</td></tr>
<tr><td>8</td><td>Reset.</td></tr>
<tr><td>9…14</td><td>Reserved.</td></tr>
<tr><td>15</td><td>EXT1/EXT2 selection. 0 = EXT1 active, 1 = EXT2 active.</td></tr>
</table></td></tr>
<tr><td>2.19</td><td>D2D REF1</td><td>FW block: D2D COMMUNICATION (page 164)</td></tr>
<tr><td></td><td colspan="2">Drive-to-drive reference 1 received through the drive-to-drive link.</td></tr>
<tr><td>2.20</td><td>D2D REF2</td><td>FW block: D2D COMMUNICATION (page 164)</td></tr>
<tr><td></td><td colspan="2">Drive-to-drive reference 2 received through the drive-to-drive link.</td></tr>
</table>

Group 03 CONTROL VALUES

3.01	SPEED REF1	FW block: SPEED REF SEL (page 110)
	Speed reference 1 in rpm.	
3.02	SPEED REF2	FW block: SPEED REF SEL (page 110)
	Speed reference 2 in rpm.	
3.03	SPEEDREF RAMP IN	FW block: SPEED REF MOD (page 111)
	Used speed reference ramp input in rpm.	
3.04	SPEEDREF RAMPED	FW block: SPEED REF RAMP (page 114)
	Ramped and shaped speed reference in rpm.	
3.05	SPEEDREF USED	FW block: SPEED ERROR (page 118)
	Used speed reference in rpm (reference before the speed error calculation).	
3.06	SPEED ERROR FILT	FW block: SPEED ERROR (page 118)
	Filtered speed error value in rpm.	
3.07	ACC COMP TORQ	FW block: SPEED ERROR (page 118)
	Output of the acceleration compensation (torque in %).	
3.08	TORQ REF SP CTRL	FW block: SPEED CONTROL (page 122)
	Limited speed controller output torque in %.	
3.09	TORQ REF1	FW block: TORQ REF SEL (page 129)
	Torque reference 1 in %.	
3.10	TORQ REF RAMPED	FW block: TORQ REF MOD (page 130)
	Ramped torque reference in %.	
3.11	TORQ REF RUSHLIM	FW block: TORQ REF MOD (page 130)
	Torque reference limited by the rush control (value in %). Torque is limited to ensure that the speed is between the defined minimum and maximum speed limits (parameters 20.01 MAXIMUM SPEED and 20.02 MINIMUM SPEED).	
3.12	TORQUE REF ADD	FW block: TORQ REF SEL (page 129)
	Torque reference additive in %.	
3.13	TORQ REF TO TC	FW block: REFERENCE CTRL (page 136)
	Torque reference in % for the torque control. When 99.05 MOTOR CTRL MODE is set to (1) Scalar, this value is forced to 0.	
3.14	BRAKE TORQ MEM	FW block: MECH BRAKE CTRL (page 139)
	Torque value (in %) stored when the mechanical brake close command is issued.	

3.15	BRAKE COMMAND	FW block: MECH BRAKE CTRL (page 139)
	Brake on/off command. 0 = Close. 1 = Open. For brake on/off control, connect this signal to a relay output (or a digital output). See section *Mechanical brake control* on page *49*.	
3.16	FLUX REF USED	FW block: MOTOR CONTROL (page 142)
	Used flux reference in percent.	
3.17	TORQUE REF USED	FW block: MOTOR CONTROL (page 142)
	Used/limited torque reference in percent.	

Group 06 DRIVE STATUS

6.01	STATUS WORD 1	FW block: DRIVE LOGIC (page 78)

Status word 1.

Bit	Name	Val.	Information
0	READY	1	Drive is ready to receive start command.
		0	Drive is not ready.
1	ENABLED	1	External run enable signal is received.
		0	No external run enable signal is received.
2	STARTED	1	Drive has received start command.
		0	Drive has not received start command.
3	RUNNING	1	Drive is modulating.
		0	Drive is not modulating.
4	EM OFF (OFF2)	1	Emergency OFF2 is active.
		0	Emergency OFF2 is inactive.
5	EM STOP (OFF3)	1	Emergency stop OFF3 (ramp stop) is active.
		0	Emergency OFF3 is inactive.
6	ACK STARTINH	1	Start inhibit is active.
		0	Start inhibit is inactive.
7	ALARM	1	An alarm is active. See chapter *Fault tracing*.
		0	No alarm
8	EXT2 ACT	1	External control EXT2 is active.
		0	External control EXT1 is active.
9	LOCAL FB	1	Fieldbus local control is active.
		0	Fieldbus local control is inactive.
10	FAULT	1	A fault is active. See chapter *Fault tracing*.
		0	No fault
11	LOCAL PANEL	1	Local control is active, i.e. drive is controlled from PC tool or control panel.
		0	Local control is inactive.
12	NOT FAULTED	1	No fault
		0	A fault is active. See chapter *Fault tracing*.
13…15	Reserved		

6.02	STATUS WORD 2	FW block: DRIVE LOGIC (page 78)

Status word 2.

Bit	Name	Val.	Information
0	START ACT	1	Drive start command is active.
		0	Drive start command is inactive.
1	STOP ACT	1	Drive stop command is active.
		0	Drive stop command is inactive.
2	READY RELAY	1	Ready to function: run enable signal on, no fault, emergency stop signal off, no ID run inhibition. Connected by default to DIO1 by par. 12.04 DIO1 OUT PTR. (Can be freely connected anywhere.)
		0	Not ready to function
3	MODULATING	1	Modulating: IGBTs are controlled, i.e. the drive is RUNNING.
		0	No modulation: IGBTs are not controlled.
4	REF RUNNING	1	Normal operation is enabled. Running. Drive follows the given reference.
		0	Normal operation is disabled, Drive is not following the given reference (e.g. in magnetisation phase drive is modulating).
5	JOGGING	1	Jogging function 1 or 2 is active.
		0	Jogging function is inactive.
6	OFF1	1	Emergency stop OFF1 is active.
		0	Emergency stop OFF1 is inactive.
7	START INH MASK	1	Maskable (by par. 10.12 START INHIBIT) start inhibit is active.
		0	No start inhibit (maskable)
8	START INH NOMASK	1	Non-maskable start inhibit is active.
		0	No start inhibit (non-maskable)
9	CHRG REL CLOSED	1	Charging relay is closed.
		0	Charging relay is open.
10	STO ACT	1	Safe Torque Off function is active. See parameter 46.07 STO DIAGNOSTIC.
		0	Safe Torque Off function is inactive.
11	Reserved		
12	RAMP IN 0	1	Ramp Function Generator input is forced to zero.
		0	Normal operation
13	RAMP HOLD	1	Ramp Function Generator output is held.
		0	Normal operation
14	RAMP OUT 0	1	Ramp Function Generator output is forced to zero.
		0	Normal operation
15	DATA LOGGER ON	1	The drive data logger is on and has not been triggered.
		0	The drive data logger is off, or its post-trigger time has not yet elapsed. See the DriveStudio user manual.

6.03 SPEED CTRL STAT — FW block: DRIVE LOGIC (page 78)

Speed control status word.

Bit	Name	Val.	Information
0	SPEED ACT NEG	1	Actual speed is negative.
1	ZERO SPEED	1	Actual speed has reached the zero speed limit (22.05 ZERO SPEED LIMIT).
2	ABOVE LIMIT	1	Actual speed has exceeded the supervision limit (22.07 ABOVE SPEED LIM).
3	AT SETPOINT	1	Difference between 1.01 SPEED ACT and 3.03 SPEEDREF RAMP IN (in speed control) or 3.05 SPEEDREF USED (in position control) is within speed window (26.07 SPEED WINDOW).
4	BAL ACTIVE	1	Speed controller output balancing is active (28.09 SPEEDCTRL BAL EN).
5	PI TUNE ACTIVE	1	Speed controller autotune is active.
6	PI TUNE REQ	1	Speed controller autotune has been requested by parameter 28.16 PI TUNE MODE.
7	PI TUNE DONE	1	Speed controller autotune has been completed successfully.
8…15	Reserved		

6.05 LIMIT WORD 1 — FW block: DRIVE LOGIC (page 78)

Limit word 1.

Bit	Name	Val.	Information
0	TORQ LIM	1	Drive torque is being limited by the motor control (undervoltage control, overvoltage control, current limitation, load angle limitation, or pull-out limitation), or by parameter 20.06 MAXIMUM TORQUE or 20.07 MINIMUM TORQUE. The source of the limitation is identified by 6.07 TORQ LIM STATUS.
1	SPD CTL TLIM MIN	1	Speed controller output minimum torque limit is active. The limit is defined by parameter 28.10 MIN TORQ SP CTRL.
2	SPD CTL TLIM MAX	1	Speed controller output maximum torque limit is active. The limit is defined by parameter 28.11 MAX TORQ SP CTRL.
3	TORQ REF MAX	1	Torque reference (3.09 TORQ REF1) maximum limit is active. The limit is defined by parameter 32.04 MAXIMUM TORQ REF.
4	TORQ REF MIN	1	Torque reference (3.09 TORQ REF1) minimum limit is active. The limit is defined by parameter 32.05 MINIMUM TORQ REF.
5	TLIM MAX SPEED	1	Torque reference maximum value is limited by the rush control, because of maximum speed limit 20.01 MAXIMUM SPEED.
6	TLIM MIN SPEED	1	Torque reference minimum value is limited by the rush control, because of minimum speed limit 20.02 MINIMUM SPEED.
7…15	Reserved		

6.07 TORQ LIM STATUS — FW block: DRIVE LOGIC (page 78)

Torque controller limitation status word.

Bit	Name	Val.	Information
0	UNDERVOLTAGE	1	Intermediate circuit DC undervoltage *
1	OVERVOLTAGE	1	Intermediate circuit DC overvoltage *
2	MINIMUM TORQUE	1	Torque reference minimum limit is active. The limit is defined by parameter 20.07 MINIMUM TORQUE. *
3	MAXIMUM TORQUE	1	Torque reference maximum limit is active. The limit is defined by parameter 20.06 MAXIMUM TORQUE. *
4	INTERNAL CURRENT	1	An inverter current limit is active. The limit is identified by bits 8…11.
5	LOAD ANGLE	1	For permanent magnet motor only: Load angle limit is active, i.e. the motor cannot produce more torque.
6	MOTOR PULLOUT	1	For asynchronous motor only: Motor pull-out limit is active, i.e. the motor cannot produce more torque.
7	Reserved		
8	THERMAL	1	Bit 4 = 0: Input current is limited by main circuit thermal limit. Bit 4 = 1: Output current is limited by main circuit thermal limit.
9	I2MAX CURRENT	1	Inverter output current limit is active. **
10	USER CURRENT	1	Maximum inverter output current limit is active. The limit is defined by parameter 20.05 MAXIMUM CURRENT. **
11…15	Reserved		

* Only one of bits 0…3 can be on simultaneously. The bit typically indicates the limit that is exceeded first.
** Only either 9 or 10 can be on simultaneously. The bit typically indicates the limit that is exceeded first.

6.12 OP MODE ACK — FW block: REFERENCE CTRL (page 136)

Operation mode acknowledge: 0 = Stopped, 1 = Speed, 2 = Torque, 3 = Min, 4 = Max, 5 = Add, 10 = Scalar, 11 = Forced magn (i.e. DC Hold).

6.14 SUPERV STATUS — FW block: SUPERVISION (page 132)

Supervision status word. See also parameter group 33 SUPERVISION (page 132).

Bit	Name	Val.	Information
0	SUPERV FUNC1 STATUS	1	Supervision function 1 is active (below low limit or over high limit)
1	SUPERV FUNC2 STATUS	1	Supervision function 2 is active (below low limit or over high limit)
2	SUPERV FUNC3 STATUS	1	Supervision function 3 is active (below low limit or over high limit)
3…15	Reserved		

Group 08 ALARMS & FAULTS

8.01	ACTIVE FAULT	FW block: FAULT FUNCTIONS (page 149)
	Fault code of the latest (active) fault.	
8.02	LAST FAULT	FW block: FAULT FUNCTIONS (page 149)
	Fault code of the 2nd latest fault.	
8.03	FAULT TIME HI	FW block: FAULT FUNCTIONS (page 149)
	Time (real time or power-on time) at which the active fault occurred in format dd.mm.yy (=day.month.year).	
8.04	FAULT TIME LO	FW block: FAULT FUNCTIONS (page 149)
	Time (real time or power-on time) at which the active fault occurred in format hh.mm.ss (hours.minutes.seconds).	
8.05	ALARM WORD 1	FW block: FAULT FUNCTIONS (page 149)

Alarm word 1. For possible causes and remedies, see chapter *Fault tracing*.

Bit	Alarm
0	BRAKE START TORQUE
1	BRAKE NOT CLOSED
2	BRAKE NOT OPEN
3	SAFE TORQUE OFF
4	STO MODE CHANGE
5	MOTOR TEMP
6	EMERGENCY OFF
7	RUN ENABLE
8	ID-RUN
9	EMERGENCY STOP
10	POSITION SCALING
11	BR OVERHEAT
12	BC OVERHEAT
13	DEVICE OVERTEMP
14	INTBOARD OVERTEMP
15	BC MOD OVERTEMP

8.06	ALARM WORD 2	FW block: FAULT FUNCTIONS (page 149)

Alarm word 2. For possible causes and remedies, see chapter *Fault tracing*.

Bit	Alarm
0	IGBT OVERTEMP
1	FIELDBUS COMM
2	LOCAL CTRL LOSS
3	AI SUPERVISION
4	Reserved
5	NO MOTOR DATA
6	ENCODER 1 FAIL
7	ENCODER 2 FAIL
8	LATCH POS 1 FAIL
9	LATCH POS 2 FAIL
10	ENC EMUL FAILURE
11	FEN TEMP FAILURE
12	ENC MAX FREQ
13	ENC REF ERROR
14	RESOLVER ERR
15	ENCODER 1 CABLE

8.07	ALARM WORD 3	FW block: FAULT FUNCTIONS (page 149)

Alarm word 3. For possible causes and remedies, see chapter *Fault tracing*.

Bit	Alarm
0	ENCODER 2 CABLE
1	D2D COMM
2	D2D BUF OVLOAD
3	PS COMM
4	RESTORE
5	CUR MEAS CALIB
6	AUTOPHASING
7	EARTH FAULT
8	Reserved
9	MOTOR NOM VALUE
10	D2D CONFIG
11…14	Reserved
15	SPEED FEEDBACK

<table>
<tr><td>8.08</td><td>ALARM WORD 4</td><td>FW block: FAULT FUNCTIONS (page 149)</td></tr>
<tr><td></td><td colspan="2">Alarm word 4. For possible causes and remedies, see chapter Fault tracing.
<table>
<tr><th>Bit</th><th>Alarm</th></tr>
<tr><td>0</td><td>OPTION COMM LOSS</td></tr>
<tr><td>1…8</td><td>Reserved</td></tr>
<tr><td>9</td><td>DC NOT CHARGED</td></tr>
<tr><td>10</td><td>SPEED TUNE FAIL</td></tr>
<tr><td>11…15</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>8.09</td><td>ALARM WORD 5</td><td>FW block: None</td></tr>
<tr><td></td><td colspan="2">Alarm word 5. For possible causes and remedies, see chapter Fault tracing.
<table>
<tr><th>Bit</th><th>Alarm</th></tr>
<tr><td>0…15</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>8.10</td><td>ALARM WORD 6</td><td>FW block: None</td></tr>
<tr><td></td><td colspan="2">Alarm word 6. For possible causes and remedies, see chapter Fault tracing.
<table>
<tr><th>Bit</th><th>Alarm</th></tr>
<tr><td>0…1</td><td>Reserved</td></tr>
<tr><td>2</td><td>LOW VOLT MOD CON</td></tr>
<tr><td>3…15</td><td>Reserved</td></tr>
</table></td></tr>
</table>

Group 09 SYSTEM INFO

9.01	DRIVE TYPE	FW block: None
	Displays the drive application type. **(1)** ACSM1 Speed: Speed and torque control application	
9.02	DRIVE RATING ID	FW block: None
	Displays the inverter type of the drive. **(0)** Unconfigured, **(1)** ACSM1-xxAx-02A5-4, **(2)** ACSM1-xxAx-03A0-4, **(3)** ACSM1-xxAx-04A0-4, **(4)** ACSM1-xxAx-05A0-4, **(5)** ACSM1-xxAx-07A0-4, **(6)** ACSM1-xxAx-09A5-4, **(7)** ACSM1-xxAx-012A-4, **(8)** ACSM1-xxAx-016A-4, **(9)** ACSM1-xxAx-024A-4, **(10)** ACSM1-xxAx-031A-4, **(11)** ACSM1-xxAx-040A-4, **(12)** ACSM1-xxAx-046A-4, **(13)** ACSM1-xxAx-060A-4, **(14)** ACSM1-xxAx-073A-4, **(15)** ACSM1-xxAx-090A-4, **(20)** ACSM1-xxAx-110A-4, **(21)** ACSM1-xxAx-135A-4, **(22)** ACSM1-xxAx-175A-4, **(23)** ACSM1-xxAx-210A-4, **(24)** ACSM1-xxCx-024A-4, **(25)** ACSM1-xxCx-031A-4, **(26)** ACSM1-xxCx-040A-4, **(27)** ACSM1-xxCx-046A-4, **(28)** ACSM1-xxCx-060A-4, **(29)** ACSM1-xxCx-073A-4, **(30)** ACSM1-xxCx-090A-4, **(31)** ACSM1-xxLx-110A-4, **(32)** ACSM1-xxLx-135A-4, **(33)** ACSM1-xxLx-175A-4, **(34)** ACSM1-xxLx-210A-4, **(35)** ACSM1-xxLx-260A-4	
9.03	FIRMWARE ID	FW block: None
	Displays the firmware name. E.g. UMFI.	
9.04	FIRMWARE VER	FW block: None
	Displays the version of the firmware package in the drive, e.g. 0x1510.	
9.05	FIRMWARE PATCH	FW block: None
	Displays the version of the firmware patch in the drive.	
9.10	INT LOGIC VER	FW block: None
	Displays the version of the logic in the power unit interface.	
9.20	OPTION SLOT 1	FW block: None
	Displays the type of the optional module in option Slot 1. **(0)** NO OPTION, **(1)** NO COMM, **(2)** UNKNOWN, **(3)** FEN-01, **(4)** FEN-11, **(5)** FEN-21, **(6)** FIO-01, **(7)** FIO-11, **(8)** FPBA-01, **(9)** FPBA-02, **(10)** FCAN-01, **(11)** FDNA-01, **(12)** FENA-01, **(13)** FENA-02, **(14)** FLON-01, **(15)** FRSA-00, **(16)** FMBA-01, **(17)** FFOA-01, **(18)** FFOA-02, **(19)** FSEN-01, **(20)** FEN-31, **(21)** FIO-21, **(22)** FSCA-01, **(23)** FSEA-21	
9.21	OPTION SLOT 2	FW block: None
	Displays the type of the optional module in option Slot 2. See 9.20 OPTION SLOT 1.	
9.22	OPTION SLOT 3	FW block: None
	Displays the type of the optional module in option Slot 3. See 9.20 OPTION SLOT 1.	

Group 10 START/STOP

Settings for

- selecting start/stop/direction signal sources for external control locations EXT1 and EXT2
- selecting sources for external fault reset, run enable and start enable signals
- selecting sources for emergency stop (OFF1 and OFF3)
- selecting source for jogging function activation signal
- enabling the start inhibit function.

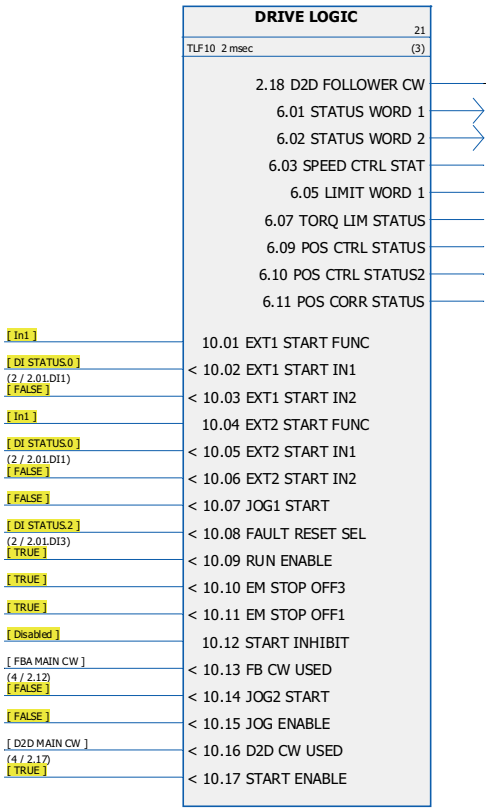
See also section *Jogging* on page 45.

Firmware block:

DRIVE LOGIC
(10)

This block

- selects the sources for the start/stop/direction signals for external control locations EXT1 and EXT2
- selects the sources for external fault reset, run enable and start enable signals
- selects the sources for emergency stop (OFF1 and OFF3)
- selects the source for jogging activation signal
- enables the start inhibit function.



<table>
<tr><td colspan="2">Block outputs located in other parameter groups</td><td>2.18 D2D FOLLOWER CW (page 67)
6.01 STATUS WORD 1 (page 70)
6.02 STATUS WORD 2 (page 71)
6.03 SPEED CTRL STAT (page 72)
6.05 LIMIT WORD 1 (page 72)
6.07 TORQ LIM STATUS (page 73)
Outputs 6.09…6.11 are not used with the Speed and Torque Control Program.</td></tr>
<tr><td>10.01</td><td>EXT1 START FUNC</td><td>FW block: DRIVE LOGIC (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source for the start and stop control in external control location EXT1.
Note: This parameter cannot be changed while the drive is running.</td></tr>
<tr><td></td><td>(0) Not sel</td><td>No source selected.</td></tr>
<tr><td></td><td>(1) In1</td><td>Source of the start and stop commands are selected by parameter 10.02 EXT1 START IN1. The start/stop is controlled as follows:<table><tr><th>Par. 10.02</th><th>Command</th></tr><tr><td>0 -> 1</td><td>Start</td></tr><tr><td>1 -> 0</td><td>Stop</td></tr></table></td></tr>
<tr><td></td><td>(2) 3-wire</td><td>Source of the start and stop commands are selected by parameters 10.02 EXT1 START IN1 and 10.03 EXT1 START IN2. The start/stop is controlled as follows:<table><tr><th>Par. 10.02</th><th>Par. 10.03</th><th>Command</th></tr><tr><td>0 -> 1</td><td>1</td><td>Start</td></tr><tr><td>Any</td><td>1 -> 0</td><td>Stop</td></tr><tr><td>Any</td><td>0</td><td>Stop</td></tr></table></td></tr>
<tr><td></td><td>(3) FBA</td><td>Start and stop control from the source selected by parameter 10.13 FB CW USED.</td></tr>
<tr><td></td><td>(4) D2D</td><td>Start and stop control from another drive via D2D Control Word.</td></tr>
<tr><td></td><td>(5) IN1 F IN2R</td><td>The source selected by 10.02 EXT1 START IN1 is the forward start signal, the source selected by 10.03 EXT1 START IN2 is the reverse start signal.<table><tr><th>Par. 10.02</th><th>Par. 10.03</th><th>Command</th></tr><tr><td>0</td><td>0</td><td>Stop</td></tr><tr><td>1</td><td>0</td><td>Start forward</td></tr><tr><td>0</td><td>1</td><td>Start reverse</td></tr><tr><td>1</td><td>1</td><td>Stop</td></tr></table></td></tr>
<tr><td></td><td>(6) IN1S IN2DIR</td><td>The source selected by 10.02 EXT1 START IN1 is the start signal (0 = stop, 1 = start), the source selected by 10.03 EXT1 START IN2 is the direction signal (0 = forward, 1 = reverse).</td></tr>
<tr><td>10.02</td><td>EXT1 START IN1</td><td>FW block: DRIVE LOGIC (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source 1 for the start and stop commands in external control location EXT1. See parameter 10.01 EXT1 START FUNC selections (1) In1 and (2) 3-wire.
Note: This parameter cannot be changed while the drive is running.</td></tr>
</table>

	Bit pointer: Group, index and bit	
10.03	EXT1 START IN2	FW block: DRIVE LOGIC (see above)
	Selects the source 2 for the start and stop commands in external control location EXT1. See parameter 10.01 EXT1 START FUNC selection (2) 3-wire. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.04	EXT2 START FUNC	FW block: DRIVE LOGIC (see above)
	Selects the source for the start and stop control in external control location EXT2. **Note:** This parameter cannot be changed while the drive is running.	
	(0) Not sel	No source selected.
	(1) In1	Source of the start and stop commands are selected by parameter 10.05 EXT2 START IN1. The start/stop is controlled as follows: **Par. 10.05** / **Command** 0 -> 1 / Start 1 -> 0 / Stop
	(2) 3-wire	Source of the start and stop commands are selected by parameters 10.05 EXT2 START IN1 and 10.06 EXT2 START IN2. The start/stop is controlled as follows: **Par. 10.05** / **Par. 10.06** / **Command** 0 -> 1 / 1 / Start Any / 1 -> 0 / Stop Any / 0 / Stop
	(3) FBA	Start and stop control from the source selected by parameter 10.13 FB CW USED.
	(4) D2D	Start and stop control from another drive via D2D Control Word.
	(5) IN1 F IN2R	The source selected by 10.05 EXT2 START IN1 is the forward start signal, the source selected by 10.06 EXT2 START IN2 is the reverse start signal. **Par. 10.05** / **Par. 10.06** / **Command** 0 / 0 / Stop 1 / 0 / Start forward 0 / 1 / Start reverse 1 / 1 / Stop
	(6) IN1S IN2DIR	The source selected by 10.05 EXT2 START IN1 is the start signal (0 = stop, 1 = start), the source selected by 10.06 EXT2 START IN2 is the direction signal (0 = forward, 1 = reverse).

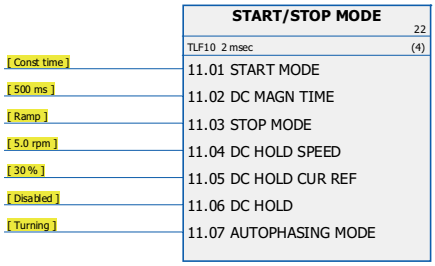
10.05	EXT2 START IN1	FW block: DRIVE LOGIC (see above)
	Selects the source 1 for the start and stop commands in external control location EXT2. See parameter 10.04 EXT2 START FUNC selections (1) In1 and (2) 3-wire. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.06	EXT2 START IN2	FW block: DRIVE LOGIC (see above)
	Selects the source 2 for the start and stop commands in external control location EXT2. See parameter 10.04 EXT2 START FUNC selection (2) 3-wire. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.07	JOG1 START	FW block: DRIVE LOGIC (see above)
	If enabled by parameter 10.15 JOG ENABLE, selects the source for the activation of jogging function 1. 1 = Active. (Jogging function 1 can also be activated through fieldbus regardless of parameter 10.15.) See section *Jogging* on page *45*. See also other jogging function parameters: 10.14 JOG2 START, 10.15 JOG ENABLE, 24.03 SPEED REF1 IN / 24.04 SPEED REF2 IN, 24.10 SPEED REF JOG1, 24.11 SPEED REF JOG2, 25.09 ACC TIME JOGGING, 25.10 DEC TIME JOGGING and 22.06 ZERO SPEED DELAY. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.08	FAULT RESET SEL	FW block: DRIVE LOGIC (see above)
	Selects the source for the external fault reset signal. The signal resets the drive after a fault trip if the cause of the fault no longer exists. 1 = Fault reset.	
	Bit pointer: Group, index and bit	
10.09	RUN ENABLE	FW block: DRIVE LOGIC (see above)
	Selects the source for the run enable signal. If the run enable signal is switched off, the drive will not start or stops if the drive is running. 1 = Run enable. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.10	EM STOP OFF3	FW block: DRIVE LOGIC (see above)
	Selects the source for the emergency stop OFF3. 0 = OFF3 active: The drive is stopped along the emergency stop ramp time, 25.11 EM STOP TIME. Emergency stop can also be activated through fieldbus (2.12 FBA MAIN CW). See section *Emergency stop* on page 53. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	

10.11	EM STOP OFF1	FW block: DRIVE LOGIC (see above)
	Selects the source for the emergency stop OFF1. 0 = OFF1 active: The drive is stopped with the active deceleration time. Emergency stop can also be activated through fieldbus (2.12 FBA MAIN CW). See section *Emergency stop* on page 53. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.12	START INHIBIT	FW block: DRIVE LOGIC (see above)
	Enables the start inhibit function. The start inhibit function prevents drive restart (i.e. protects against unexpected start) if • drive trips on a fault and fault is reset. • run enable signal activates while the start command is active. See parameter 10.09 RUN ENABLE. • control changes from local to remote. • external control switches from EXT1 to EXT2 or from EXT2 to EXT1. An active start inhibit can be reset with a stop command. Note that in certain applications it is necessary to allow the drive to restart.	
	(0) Disabled	Start inhibit function disabled.
	(1) Enabled	Start inhibit function enabled.
10.13	FB CW USED	FW block: DRIVE LOGIC (see above)
	Selects the source for the control word when fieldbus (FBA) is selected as the external start and stop control location (see parameters 10.01 EXT1 START FUNC and 10.04 EXT2 START FUNC). By default, the source is parameter 2.12 FBA MAIN CW. **Note:** This parameter cannot be changed while the drive is running.	
	Value pointer: Group and index	
10.14	JOG2 START	FW block: DRIVE LOGIC (see above)
	If enabled by parameter 10.15 JOG ENABLE, selects the source for the activation of jogging function 2. 1 = Active. (Jogging function 2 can also be activated through fieldbus regardless of parameter 10.15.) **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
10.15	JOG ENABLE	FW block: DRIVE LOGIC (see above)
	Selects the source for enabling parameters 10.07 JOG1 START and 10.14 JOG2 START. **Note:** Jogging can be enabled using this parameter only when no start command from an external control location is active. On the other hand, if jogging is already enabled, the drive cannot be started from an external control location apart from jog commands through fieldbus.	
	Bit pointer: Group, index and bit	
10.16	D2D CW USED	FW block: DRIVE LOGIC (see above)
	Selects the source for the control word for drive-to-drive communication. By default, the source is parameter 2.17 D2D MAIN CW.	

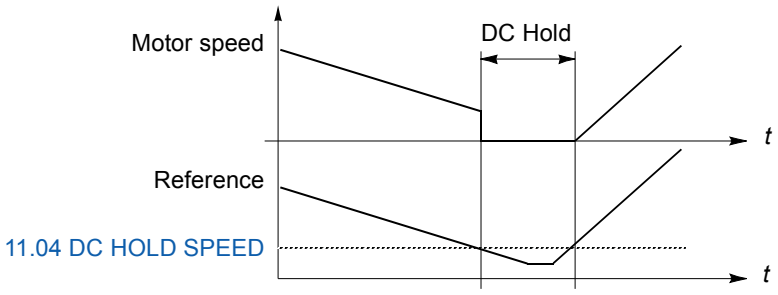
	Value pointer: Group and index	
10.17	START ENABLE	FW block: DRIVE LOGIC (see above)
	Selects the source for the start enable signal. If the start enable signal is switched off, the drive will not start or stops if the drive is running. 1 = Start enable. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	

Group 11 START/STOP MODE

These parameters select the start and stop functions as well as the autophasing mode, define the DC magnetising time of the motor, and configure the DC hold function.

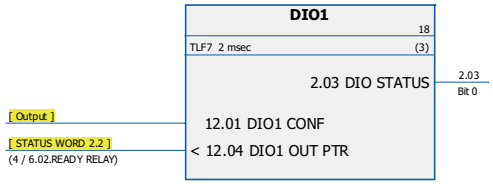
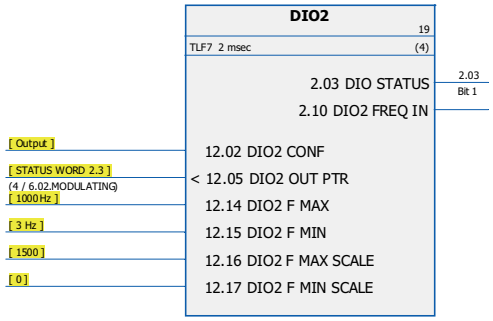
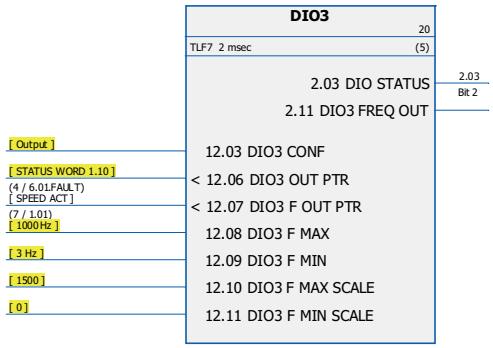
<table>
<tr><td colspan="2">Firmware block:
START/STOP MODE
(11)</td><td> 
</td></tr>
<tr><td>11.01</td><td>START MODE</td><td>FW block: START/STOP MODE (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the motor start function.
Notes:
• This parameter has no effect if parameter 99.05 MOTOR CTRL MODE is set to (1) Scalar.
• Starting to a rotating machine is not possible when DC magnetising is selected ((0) Fast or (1) Const time).
• With permanent magnet motors, automatic start must be used.
• This parameter cannot be changed while the drive is running.</td></tr>
<tr><td></td><td>(0) Fast</td><td>DC magnetising should be selected if a high break-away torque is required. The drive pre-magnetises the motor before the start. The pre-magnetising time is determined automatically, being typically 200 ms to 2 s depending on the motor size.</td></tr>
<tr><td></td><td>(1) Const time</td><td>Constant DC magnetising should be selected instead of FAST DC magnetising if a constant pre-magnetising time is required (e.g. if the motor start must be simultaneous with a mechanical brake release). This selection also guarantees the highest possible break-away torque when the pre-magnetising time is set long enough. The pre-magnetising time is defined by parameter 11.02 DC MAGN TIME.
WARNING! The drive will start after the set magnetising time has passed even if motor magnetisation is not completed. In applications where a full break-away torque is essential, ensure that the constant magnetising time is long enough to allow generation of full magnetisation and torque.</td></tr>
<tr><td></td><td>(2) Automatic</td><td>Automatic start guarantees optimal motor start in most cases. It includes the flying start function (starting to a rotating machine) and the automatic restart function (stopped motor can be restarted immediately without waiting the motor flux to die away). The drive motor control program identifies the flux as well as the mechanical state of the motor and starts the motor instantly under all conditions.</td></tr>
</table>

<table>
<tr><td>11.02</td><td>DC MAGN TIME</td><td>FW block: START/STOP MODE (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the constant DC magnetising time. See parameter 11.01 START MODE. After the start command, the drive automatically premagnetises the motor the set time.
To ensure full magnetising, set this value to the same value as or higher than the rotor time constant. If not known, use the rule-of-thumb value given in the table below:
<table><tr><th>Motor rated power</th><th>Constant magnetising time</th></tr><tr><td>< 1 kW</td><td>≥ 50 to 100 ms</td></tr><tr><td>1 to 10 kW</td><td>≥ 100 to 200 ms</td></tr><tr><td>10 to 200 kW</td><td>≥ 200 to 1000 ms</td></tr><tr><td>200 to 1000 kW</td><td>≥ 1000 to 2000 ms</td></tr></table>
Note: This parameter cannot be changed while the drive is running.</td></tr>
<tr><td></td><td>0…10000 ms</td><td>DC magnetising time.</td></tr>
<tr><td>11.03</td><td>STOP MODE</td><td>FW block: START/STOP MODE (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the motor stop function.</td></tr>
<tr><td></td><td>(1) Coast</td><td>Stop by cutting of the motor power supply. The motor coasts to a stop.
WARNING! If the mechanical brake is used, ensure it is safe to stop the drive by coasting. For more information on mechanical brake function, see parameter group 35 MECH BRAKE CTRL.</td></tr>
<tr><td></td><td>(2) Ramp</td><td>Stop along ramp. See parameter group 25 SPEED REF RAMP.</td></tr>
<tr><td>11.04</td><td>DC HOLD SPEED</td><td>FW block: START/STOP MODE (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the DC hold speed. See parameter 11.06 DC HOLD.</td></tr>
<tr><td></td><td>0…1000 rpm</td><td>DC hold speed.</td></tr>
<tr><td>11.05</td><td>DC HOLD CUR REF</td><td>FW block: START/STOP MODE (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the DC hold current in percent of the motor nominal current. See parameter 11.06 DC HOLD.</td></tr>
<tr><td></td><td>0…100%</td><td>DC hold current.</td></tr>
</table>

11.06	DC HOLD	FW block: START/STOP MODE (see above)
	Enables the DC hold function. The function makes it possible to lock the rotor at zero speed. When both the reference and the speed drop below the value of parameter 11.04 DC HOLD SPEED, the drive will stop generating sinusoidal current and start to inject DC into the motor. The current is set by parameter 11.05 DC HOLD CUR REF. When the reference speed exceeds parameter 11.04 DC HOLD SPEED, normal drive operation continues.  **Notes:** • The DC hold function has no effect if the start signal is switched off. • The DC hold function can only be activated in speed control mode. • The DC hold function cannot be activated if par. 99.05 MOTOR CTRL MODE is set to (1) Scalar. • Injecting DC current into the motor causes the motor to heat up. In applications where long DC hold times are required, externally ventilated motors should be used. If the DC hold period is long, the DC hold cannot prevent the motor shaft from rotating if a constant load is applied to the motor.	
	(0) Disabled	DC hold function disabled.
	(1) Enabled	DC hold function enabled.
11.07	AUTOPHASING MODE	FW block: START/STOP MODE (see above)
	Selects the way the autophasing routine is performed. See also section *Autophasing* on page 38.	
	(0) Turning	This mode gives the most accurate autophasing result. This mode can be used, and is recommended if it is allowed for the motor to rotate and the start-up is not time-critical. **Note:** This mode will cause the motor to rotate during the ID run.
	(1) Standstill 1	Faster than the (0) Turning mode, but not as accurate. The motor will not rotate.
	(2) Standstill 2	An alternative standstill autophasing mode that can be used if the TURNING mode cannot be used, and the (1) Standstill 1 mode gives erratic results. However, this mode is considerably slower than (1) Standstill 1.

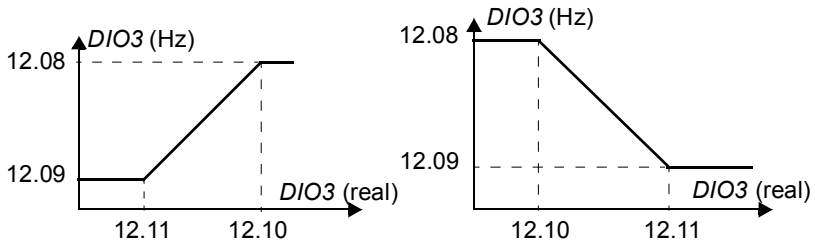
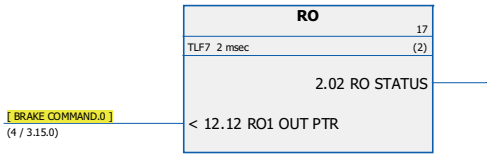
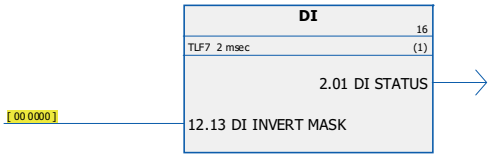
Group 12 DIGITAL IO

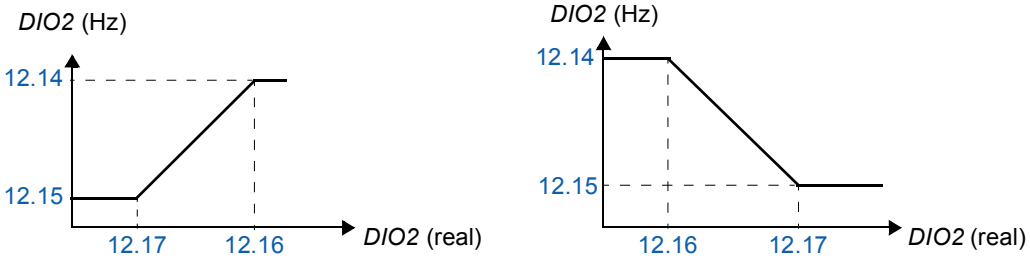
Settings for the digital inputs and outputs, and the relay output.

Firmware block: **DIO1** (6) Selects whether DIO1 is used as a digital input or as a digital output and connects an actual signal to the digital output. The block also shows the DIO status.	
Block outputs located in other parameter groups	2.03 DIO STATUS (page 62)
Firmware block: **DIO2** (7) Selects whether DIO2 is used as a digital or frequency input or as a digital output and connects an actual signal to the digital output. The block also shows the DIO status. Frequency input can be scaled with standard function blocks. See chapter *Standard function blocks*.	
Block outputs located in other parameter groups	2.03 DIO STATUS (page 62) 2.10 DIO2 FREQ IN (page 62)
Firmware block: **DIO3** (8) Selects whether DIO3 is used as a digital input or as a digital/frequency output, connects an actual signal to the digital/frequency output and scales the frequency output. The block also shows the DIO status.	
Block outputs located in other parameter groups	2.03 DIO STATUS (page 62) 2.11 DIO3 FREQ OUT (page 62)

12.01	DIO1 CONF	FW block: DIO1 (see above)
	Selects whether DIO1 is used as a digital input or as a digital output.	
	(0) Output	DIO1 is used as a digital output.
	(1) Input	DIO1 is used as a digital input.

12.02	DIO2 CONF	FW block: DIO2 (see above)
	Selects whether DIO2 is used as a digital input, as a digital output or as a frequency input.	
	(0) Output	DIO2 is used as a digital output.
	(1) Input	DIO2 is used as a digital input.
	(2) Freq input	DIO2 is used as a frequency input.
12.03	DIO3 CONF	FW block: DIO3 (see above)
	Selects whether DIO3 is used as a digital input, as a digital output or as a frequency output.	
	(0) Output	DIO2 is used as a digital output.
	(1) Input	DIO2 is used as a digital input.
	(2) Freq output	DIO2 is used as a frequency output.
12.04	DIO1 OUT PTR	FW block: DIO1 (see above)
	Selects a drive signal to be connected to digital output DIO1 (when 12.01 DIO1 CONF is set to (0) Output).	
	Bit pointer: Group, index and bit	
12.05	DIO2 OUT PTR	FW block: DIO2 (see above)
	Selects a drive signal to be connected to digital output DIO2 (when 12.02 DIO2 CONF is set to (0) Output).	
	Bit pointer: Group, index and bit	
12.06	DIO3 OUT PTR	FW block: DIO3 (see above)
	Selects a drive signal to be connected to digital output DIO3 (when 12.03 DIO3 CONF is set to (0) Output).	
	Bit pointer: Group, index and bit	
12.07	DIO3 F OUT PTR	FW block: DIO3 (see above)
	Selects a drive signal to be connected to frequency output (when 12.03 DIO3 CONF is set to (2) Freq output).	
	Value pointer: Group and index	
12.08	DIO3 F MAX	FW block: DIO3 (see above)
	Defines the maximum value for frequency output (when 12.03 DIO3 CONF is set to (2) Freq output).	
	3…32768 Hz	Maximum DIO3 output frequency.
12.09	DIO3 F MIN	FW block: DIO3 (see above)
	Defines the minimum value for frequency output (when 12.03 DIO3 CONF is set to (2) Freq output).	
	3…32768 Hz	Minimum DIO3 output frequency.

12.10	DIO3 F MAX SCALE	FW block: DIO3 (see above)
	Defines the real value that corresponds to the maximum frequency output value defined by parameter 12.08 DIO3 F MAX. 	
	0…32768	Real value corresponding to value of parameter 12.08.
12.11	DIO3 F MIN SCALE	FW block: DIO3 (see above)
	Defines the real value that corresponds to the minimum frequency output value defined by parameter 12.09 DIO3 F MIN. See parameter 12.10 DIO3 F MAX SCALE.	
	0…32768	Real value corresponding to value of parameter 12.09.
Firmware block: **RO** (5) Connects an actual signal to the relay output. The block also shows the relay output status.		
Block outputs located in other parameter groups		2.02 RO STATUS (page 62)
12.12	RO1 OUT PTR	FW block: RO (see above)
	Selects a drive signal to be connected to relay output RO1.	
	Bit pointer: Group, index and bit	
Firmware block: **DI** (4) Shows the status of the digital inputs. Inverts the status of any DI if desired.		
Block outputs located in other parameter groups		2.01 DI STATUS (page 62)
12.13	DI INVERT MASK	FW block: DI (see above)
	Inverts status of digital inputs as reported by 2.01 DI STATUS. For example, a value of 0b000100 inverts the status of DI3 in the signal.	
	0b000000…0b111111	DI status inversion mask.

12.14	DIO2 F MAX	FW block: DIO2 (see above)
	Defines the maximum value for frequency input (when 12.02 DIO2 CONF is set to (2) Freq input).	
	3…32768 Hz	Maximum DIO2 input frequency.
12.15	DIO2 F MIN	FW block: DIO2 (see above)
	Defines the minimum value for frequency input (when 12.02 DIO2 CONF is set to (2) Freq input).	
	3…32768 Hz	Minimum DIO2 input frequency.
12.16	DIO2 F MAX SCALE	FW block: DIO2 (see above)
	Defines the real value that corresponds to the maximum frequency input value defined by parameter 12.14 DIO2 F MAX. 	
	-32768…32768	Real value corresponding to value of parameter 12.14.
12.17	DIO2 F MIN SCALE	FW block: DIO2 (see above)
	Defines the real value that corresponds to the minimum frequency input value defined by parameter 12.15 DIO2 F MIN. See parameter 12.16 DIO2 F MAX SCALE.	
	-32768…32768	Real value corresponding to value of parameter 12.15.

Group 13 ANALOGUE INPUTS

Settings for the analogue inputs.

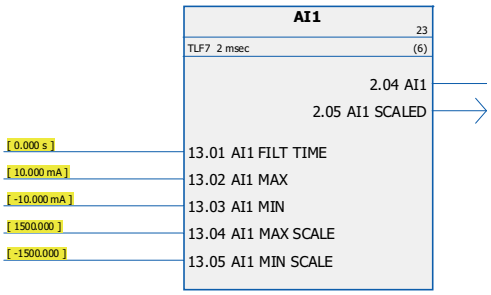
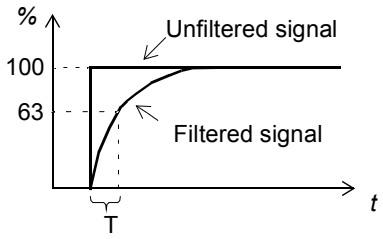
The drive offers two programmable analogue inputs, AI1 and AI2. Both inputs can be used either as a voltage or a current input (-11…11 V or -22…22 mA). The input type is selected with jumpers J1 and J2 respectively on the JCU Control Unit.

The inaccuracy of the analogue inputs is 1% of the full scale range and the resolution is 11 bits (+ sign). The hardware filter time constant is approximately 0.25 ms.

Analogue inputs can be used as the source for speed and torque reference.

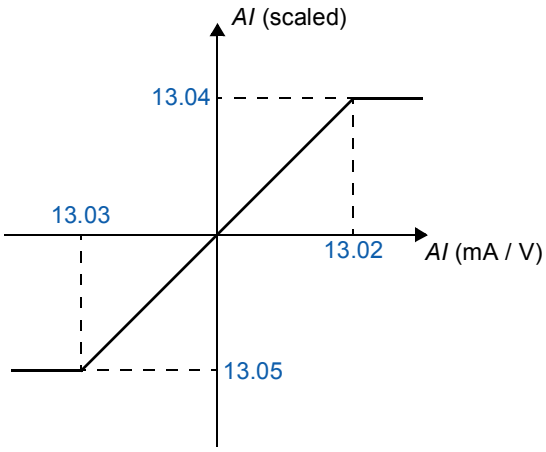
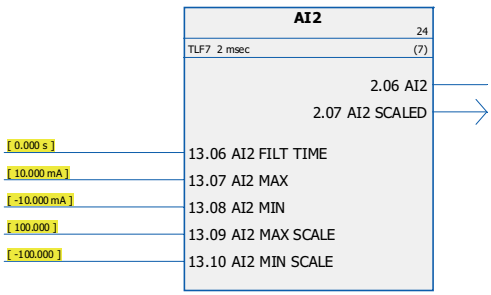
Analogue input supervision can be added with standard function blocks. See chapter *Standard function blocks*.

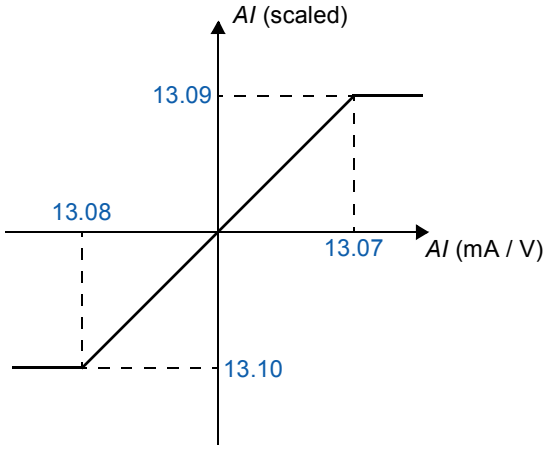
<table>
<tr><td>Firmware block:
AI1
(12)

Filters and scales the analogue input AI1 signal and selects the AI1 supervision. Also shows the value of the input.</td><td colspan="2"> 
</td></tr>
<tr><td>Block outputs located in other parameter groups</td><td colspan="2">2.04 AI1 (page 62)
2.05 AI1 SCALED (page 62)</td></tr>
<tr><td>13.01</td><td>AI1 FILT TIME</td><td>FW block: AI1 (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the filter time constant for analogue input AI1.
 

$O = I \cdot (1 - e^{-t/T})$
I = filter input (step)
O = filter output
t = time
T = filter time constant

Note: The signal is also filtered due to the signal interface hardware (approximately 0.25 ms time constant). This cannot be changed by any parameter.</td></tr>
<tr><td></td><td>0…30 s</td><td>Filter time constant for AI1.</td></tr>
<tr><td>13.02</td><td>AI1 MAX</td><td>FW block: AI1 (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the maximum value for analogue input AI1. The type is selected with jumper J1 on the JCU Control Unit.</td></tr>
<tr><td></td><td>-11…11 V / -22…22 mA</td><td>Maximum AI1 input value.</td></tr>
</table>

13.03	AI1 MIN	FW block: AI1 (see above)
	Defines the minimum value for analogue input AI1. The type is selected with jumper J1 on the JCU Control Unit.	
	-11…11 V / -22…22 mA	Minimum AI1 input value.
13.04	AI1 MAX SCALE	FW block: AI1 (see above)
	Defines the real value that corresponds to the maximum analogue input value defined by parameter 13.02 AI1 MAX.  	
	-32768…32768	Real value corresponding to value of parameter 13.02.
13.05	AI1 MIN SCALE	FW block: AI1 (see above)
	Defines the real value that corresponds to the minimum analogue input value defined by parameter 13.03 AI1 MIN. See parameter 13.04 AI1 MAX SCALE.	
	-32768…32768	Real value corresponding to value of parameter 13.03.
Firmware block: **AI2** (13) Filters and scales the analogue input AI2 signal and selects the AI2 supervision. Also shows the value of the input.		
Block outputs located in other parameter groups		2.06 AI2 (page 62) 2.07 AI2 SCALED (page 62)
13.06	AI2 FILT TIME	FW block: AI2 (see above)
	Defines the filter time constant for analogue input AI2. See parameter 13.01 AI1 FILT TIME.	
	0…30 s	Filter time constant for AI2.

13.07	AI2 MAX	FW block: AI2 (see above)
	Defines the maximum value for analogue input AI2. The type is selected with jumper J2 on the JCU Control Unit.	
	-11…11 V / -22…22 mA	Maximum AI2 input value.
13.08	AI2 MIN	FW block: AI2 (see above)
	Defines the minimum value for analogue input AI2. The type is selected with jumper J2 on the JCU Control Unit.	
	-11…11 V / -22…22 mA	Minimum AI2 input value.
13.09	AI2 MAX SCALE	FW block: AI2 (see above)
	Defines the real value that corresponds to the maximum analogue input value defined by parameter 13.07 AI2 MAX.  	
	-32768…32768	Real value corresponding to value of parameter 13.07.
13.10	AI2 MIN SCALE	FW block: AI2 (see above)
	Defines the real value that corresponds to the minimum analogue input value defined by parameter 13.08 AI2 MIN. See parameter 13.09 AI2 MAX SCALE.	
	-32768…32768	Real value corresponding to value of parameter 13.08.
13.11	AITUNE	FW block: None
	Triggers the AI tuning function. Connect the signal to the input and select the appropriate tuning function.	
	(0) No action	AI tune is not activated.
	(1) AI1 min tune	Current analogue input AI1 signal value is set as minimum value for AI1, parameter 13.03 AI1 MIN. The value reverts back to (0) No action automatically.
	(2) AI1 max tune	Current analogue input AI1 signal value is set as maximum value for AI1, parameter 13.02 AI1 MAX. The value reverts back to (0) No action automatically.

	(3) AI2 min tune	Current analogue input AI2 signal value is set as minimum value for AI2, parameter 13.08 AI2 MIN. The value reverts back to (0) No action automatically.
	(4) AI2 max tune	Current analogue input AI2 signal value is set as maximum value for AI2, parameter 13.07 AI2 MAX. The value reverts back to (0) No action automatically.
13.12	AI SUPERVISION	FW block: None
	Selects how the drive reacts when analogue input signal limit is reached. The limit is selected by parameter 13.13 AI SUPERVIS ACT.	
	(0) No	No action taken.
	(1) Fault	The drive trips on fault AI SUPERVISION.
	(2) Spd ref Safe	The drive generates alarm AI SUPERVISION and sets the speed to the speed defined by parameter 46.02 SPEED REF SAFE. **WARNING!** Make sure that it is safe to continue operation in case of a communication break.
	(3) Last speed	The drive generates alarm AI SUPERVISION and freezes the speed to the level the drive was operating at. The speed is determined by the average speed over the previous 10 seconds. **WARNING!** Make sure that it is safe to continue operation in case of a communication break.
13.13	AI SUPERVIS ACT	FW block: None
	Selects the analogue input signal supervision limit. **Bit** | **Supervision selected by parameter 13.12 AI SUPERVISION is activated if** 0 | AI1<min | AI1 signal value falls below the value defined by equation: par. 13.03 AI1 MIN - 0.5 mA or V 1 | AI1>max | AI1 signal value exceeds the value defined by equation: par. 13.02 AI1 MAX + 0.5 mA or V 2 | AI2<min | AI2 signal value falls below the value defined by equation: par. 13.08 AI2 MIN - 0.5 mA or V 3 | AI2>max | AI2 signal value exceeds the value defined by equation: par. 13.07 AI2 MAX + 0.5 mA or V Example: If parameter value is set to 0010 (bin), bit 1 AI1>max is selected.	
	0b0000…0b1111	AI1/AI2 signal supervision selection.

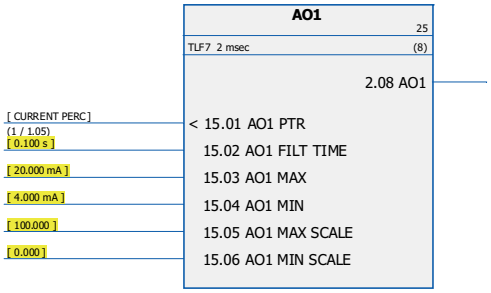
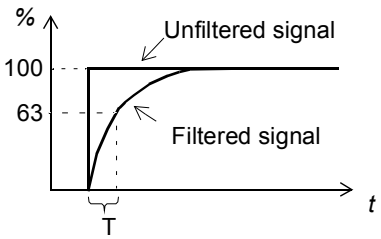
Group 15 ANALOGUE OUTPUTS

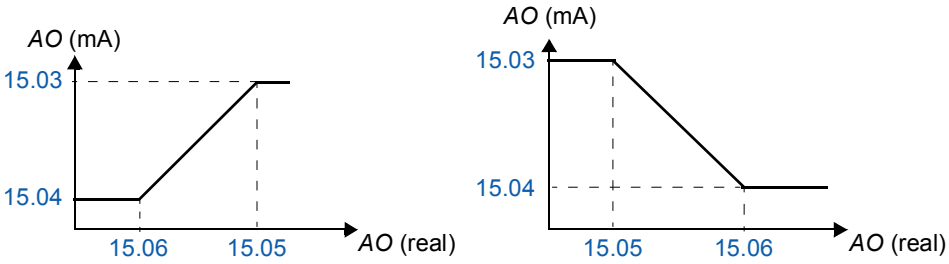
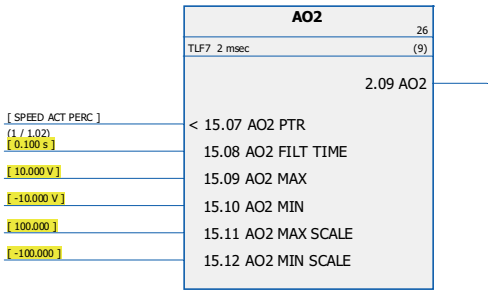
Settings for the analogue outputs.

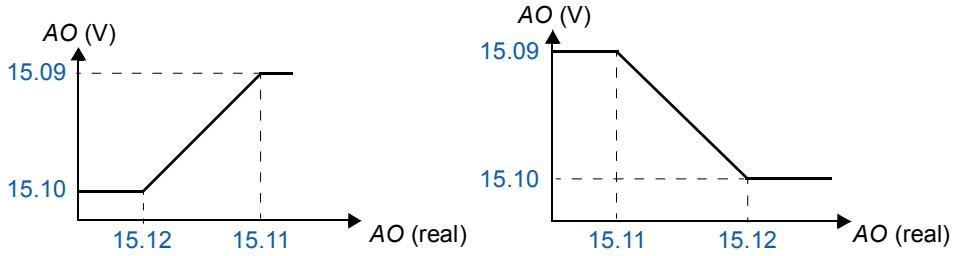
The drive offers two programmable analogue outputs: one current output AO1 (0…20 mA) and one voltage output AO2 (-10…10 V).

The resolution of the analogue outputs is 11 bits (+ sign) and the inaccuracy is 2% of the full scale range.

The analogue output signals can be proportional to motor speed, process speed (scaled motor speed), output frequency, output current, motor torque, motor power, etc. It is possible to write a value to an analogue output through a serial communication link (e.g. fieldbus link).

Firmware block: **AO1** (14) Connects an actual signal to analogue output AO1, and filters and scales the output signal. Also shows the value of the output.		
Block outputs located in other parameter groups		2.08 AO1 (page 62)
15.01	AO1 PTR	FW block: AO1 (see above)
	Selects a drive signal to be connected to analogue output AO1.	
	Value pointer: Group and index	
15.02	AO1 FILT TIME	FW block: AO1 (see above)
	Defines the filtering time constant for analogue output AO1.  $O = I \cdot (1 - e^{-t/T})$ I = filter input (step) O = filter output t = time T = filter time constant **Note:** The signal is also filtered due to the signal interface hardware (approximately 0.5 ms time constant). This cannot be changed by any parameter.	
	0…30 s	Filter time constant for AO1.
15.03	AO1 MAX	FW block: AO1 (see above)
	Defines the maximum value for analogue output AO1.	
	0…22.7 mA	Maximum AO1 output value.

15.04	AO1 MIN	FW block: AO1 (see above)
	Defines the minimum value for analogue output AO1.	
	0…22.7 mA	Minimum AO1 output value.
15.05	AO1 MAX SCALE	FW block: AO1 (see above)
	Defines the real value that corresponds to the maximum analogue output value defined by parameter 15.03 AO1 MAX. 	
	-32768…32767	Real value corresponding to value of parameter 15.03.
15.06	AO1 MIN SCALE	FW block: AO1 (see above)
	Defines the real value that corresponds to the minimum analogue output value defined by parameter 15.04 AO1 MIN. See parameter 15.05 AO1 MAX SCALE.	
	-32768…32767	Real value corresponding to value of parameter 15.04.
Firmware block: **AO2** (15) Connects an actual signal to analogue output AO2, and filters and scales the output signal. Also shows the value of the output.		
Block outputs located in other parameter groups		2.09 AO2 (page 62)
15.07	AO2 PTR	FW block: AO2 (see above)
	Selects a drive signal to be connected to analogue output AO2.	
	Value pointer: Group and index	
15.08	AO2 FILT TIME	FW block: AO2 (see above)
	Defines the filtering time constant for analogue output AO2. See parameter 15.02 AO1 FILT TIME.	
	0…30 s	Filter time constant for AO2.
15.09	AO2 MAX	FW block: AO2 (see above)
	Defines the maximum value for analogue output AO2.	

	-10…10 V	Maximum AO2 output value.
15.10	AO2 MIN	FW block: AO2 (see above)
	Defines the minimum value for analogue output AO2.	
	-10…10 V	Minimum AO2 output value.
15.11	AO2 MAX SCALE	FW block: AO2 (see above)
	Defines the real value that corresponds to the maximum analogue output value defined by parameter 15.09 AO2 MAX. 	
	-32768…32767	Real value corresponding to value of parameter 15.09.
15.12	AO2 MIN SCALE	FW block: AO2 (see above)
	Defines the real value that corresponds to the minimum analogue output value defined by parameter 15.10 AO2 MIN. See parameter 15.11 AO2 MAX SCALE.	
	-32768…32767	Real value corresponding to value of parameter 15.10.

Group 16 SYSTEM

Local control and parameter access settings, restoration of default parameter values, save of parameters into permanent memory.

16.01	LOCAL LOCK	FW block: None
	Selects the source for disabling local control (Take/Release button on the PC tool, LOC/REM key of the panel). 1 = Local control disabled. 0 = Local control enabled. **WARNING!** Before activating, ensure that the control panel is not needed for stopping the drive!	
	Bit pointer: Group, index and bit	
16.02	PARAMETER LOCK	FW block: None
	Selects the state of the parameter lock. The lock prevents parameter changing. **Note:** This parameter can only be adjusted after the correct pass code has been entered at parameter 16.03 PASS CODE.	
	(0) Locked	Locked. Parameter values cannot be changed from the control panel.
	(1) Open	The lock is open. Parameter values can be changed.
	(2) Not saved	The lock is open. Parameter values can be changed, but the changes will not be stored at power switch off.
16.03	PASS CODE	FW block: None
	After entering 358 at this parameter, parameter 16.02 PARAMETER LOCK can be adjusted. The value reverts back to 0 automatically.	
16.04	PARAM RESTORE	FW block: None
	Restores the original settings of the application, i.e. parameter factory default values. **Note:** This parameter cannot be changed while the drive is running.	
	(0) Done	Restoration is completed.
	(1) Restore defs	All parameter values are restored to default values, except motor data, ID run results, and fieldbus, drive-to-drive link and encoder configuration data.
	(2) Clear all	All parameter values are restored to default values, including motor data, ID run results and fieldbus and encoder configuration data. PC tool communication is interrupted during the restoration. Drive CPU is re-booted after the restoration is completed.
16.07	PARAM SAVE	FW block: None
	Saves the valid parameter values to permanent memory. See also section *Programming via parameters* on page 30.	
	(0) Done	Save completed.
	(1) Save	Save in progress.

16.09	USER SET SEL	FW block: None
	Enables the save and restoration of up to four custom sets of parameter settings. The set that was in use before powering down the drive is in use after the next power-up. **Note:** Any parameter changes made after loading a user set are not automatically stored into the loaded set – they must be saved using this parameter.	
	(1) No request	Load or save operation complete; normal operation.
	(2) Load set 1	Load user parameter set 1.
	(3) Load set 2	Load user parameter set 2.
	(4) Load set 3	Load user parameter set 3.
	(5) Load set 4	Load user parameter set 4.
	(6) Save set 1	Save user parameter set 1.
	(7) Save set 2	Save user parameter set 2.
	(8) Save set 3	Save user parameter set 3.
	(9) Save set 4	Save user parameter set 4.
	(10) IO mode	Load user parameter set using parameters 16.11 and 16.12.
16.10	USER SET LOG	FW block: None
	Shows the status of the user parameter sets (see parameter 16.09 USER SET SEL). Read-only.	
	N/A	No user sets have been saved.
	(1) Loading	A user set is being loaded.
	(2) Saving	A user set is being saved.
	(4) Faulted	Invalid or empty parameter set.
	(8) Set1 IO act	User parameter set 1 has been selected by parameters 16.11 and 16.12.
	(16) Set2 IO act	User parameter set 2 has been selected by parameters 16.11 and 16.12.
	(32) Set3 IO act	User parameter set 3 has been selected by parameters 16.11 and 16.12.
	(64) Set4 IO act	User parameter set 4 has been selected by parameters 16.11 and 16.12.
	(128) Set1 par act	User parameter set 1 has been loaded using parameter 16.09.
	(256) Set2 par act	User parameter set 2 has been loaded using parameter 16.09.
	(512) Set3 par act	User parameter set 3 has been loaded using parameter 16.09.
	(1024) Set4 par act	User parameter set 4 has been loaded using parameter 16.09.

<table>
<tr><td>16.11</td><td>USER IO SET LO</td><td>FW block: None</td></tr>
<tr><td></td><td colspan="2">Together with parameter 16.12 USER IO SET HI, selects the user parameter set when parameter 16.09 USER SET SEL is set to (10) IO mode. The status of the source defined by this parameter and parameter 16.12 select the user parameter set as follows:
<table>
<tr><th>Status of source defined by par. 16.11</th><th>Status of source defined by par. 16.12</th><th>User parameter set selected</th></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Set 1</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Set 2</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Set 3</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>Set 4</td></tr>
</table></td></tr>
<tr><td></td><td colspan="2">Bit pointer: Group, index and bit</td></tr>
<tr><td>16.12</td><td>USER IO SET HI</td><td>FW block: None</td></tr>
<tr><td></td><td colspan="2">See parameter 16.11 USER IO SET LO.</td></tr>
<tr><td></td><td colspan="2">Bit pointer: Group, index and bit</td></tr>
<tr><td>16.13</td><td>TIME SOURCE PRIO</td><td>FW block: None</td></tr>
<tr><td></td><td colspan="2">Selects which real-time clock source is adopted by the drive as the master real-time clock. Some selections specify multiple sources that are in order of priority.</td></tr>
<tr><td></td><td>(0) FB_D2D_MMI</td><td>Fieldbus (highest priority); drive-to-drive link; man-machine interface (control panel or PC).</td></tr>
<tr><td></td><td>(1) D2D_FB_MMI</td><td>Drive-to-drive link (highest priority); fieldbus; man-machine interface (control panel or PC).</td></tr>
<tr><td></td><td>(2) FB_D2D</td><td>Fieldbus (highest priority); drive-to-drive link.</td></tr>
<tr><td></td><td>(3) D2D_FB</td><td>Drive-to-drive link (highest priority); fieldbus.</td></tr>
<tr><td></td><td>(4) FB Only</td><td>Fieldbus only.</td></tr>
<tr><td></td><td>(5) D2D Only</td><td>Drive-to-drive link only.</td></tr>
<tr><td></td><td>(6) MMI_FB_D2D</td><td>Man-machine interface (control panel or PC) (highest priority); fieldbus; drive-to-drive link.</td></tr>
<tr><td></td><td>(7) MMI Only</td><td>Man-machine interface (control panel or PC) only.</td></tr>
<tr><td></td><td>(8) Internal</td><td>No external sources are used as master real-time clock.</td></tr>
</table>

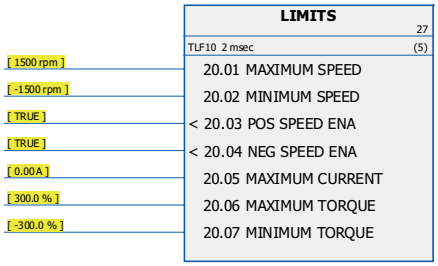
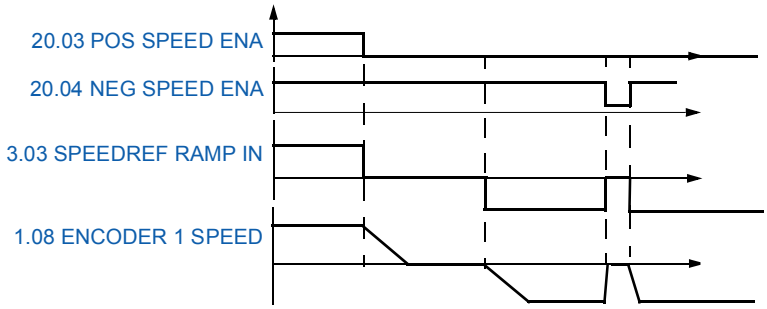
Group 17 PANEL DISPLAY

Selection of signals for panel display.

17.01	SIGNAL1 PARAM	FW block: None
	Selects the first signal to be displayed on the control panel. The default signal is 1.03 FREQUENCY.	
	Value pointer: Group and index	
17.02	SIGNAL2 PARAM	FW block: None
	Selects the second signal to be displayed on the control panel. The default signal is 1.04 CURRENT.	
	Value pointer: Group and index	
17.03	SIGNAL3 PARAM	FW block: None
	Selects the third signal to be displayed on the control panel. The default signal is 1.06 TORQUE.	
	Value pointer: Group and index	

Group 20 LIMITS

Definition of drive operation limits.

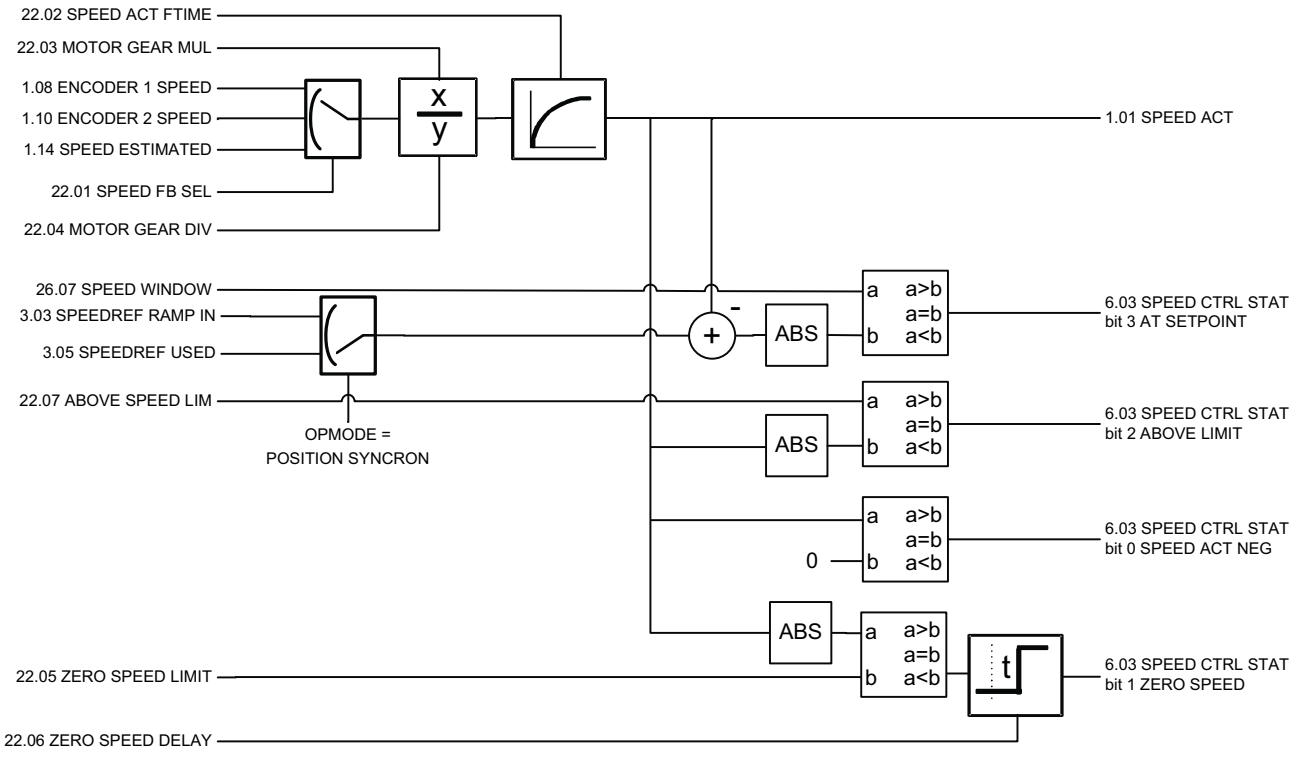
Firmware block: **LIMITS** (20) Adjusts the drive speed, current and torque limits, selects the source for the positive/negative speed reference enable command and enables the thermal current limitation.		
20.01	MAXIMUM SPEED	FW block: LIMITS (see above).
	Defines the allowed maximum speed. See also parameter 22.08 SPEED TRIPMARGIN.	
	0…30000 rpm	Allowed maximum speed.
20.02	MINIMUM SPEED	FW block: LIMITS (see above).
	Defines the allowed minimum speed. See also parameter 22.08 SPEED TRIPMARGIN.	
	-30000…0 rpm	Allowed minimum speed.
20.03	POS SPEED ENA	FW block: LIMITS (see above).
	Selects the source of the positive speed reference enable command. 1 = Positive speed reference is enabled. 0 = Positive speed reference is interpreted as zero speed reference (In the figure below 3.03 SPEEDREF RAMP IN is set to zero after the positive speed enable signal has cleared). Actions in different control modes: Speed control: Speed reference is set to zero and the motor is stopped along the currently active deceleration ramp. Torque control: Torque limit is set to zero and the rush controller stops the motor.  **Example:** The motor is rotating in the forward direction. To stop the motor, the positive speed enable signal is deactivated by a hardware limit switch (e.g. via digital input). If the positive speed enable signal remains deactivated and the negative speed enable signal is active, only reverse rotation of the motor is allowed.	
	Bit pointer: Group, index and bit	

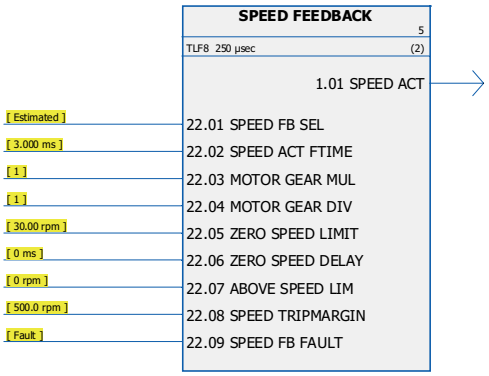
20.04	NEG SPEED ENA	FW block: LIMITS (see above).
	Selects the source of the negative speed reference enable command. See parameter 20.03 POS SPEED ENA.	
	Bit pointer: Group, index and bit	
20.05	MAXIMUM CURRENT	FW block: LIMITS (see above).
	Defines the allowed maximum motor current.	
	0…30000 A	Maximum allowed motor current.
20.06	MAXIMUM TORQUE	FW block: LIMITS (see above).
	Defines the maximum torque limit for the drive (in percent of the motor nominal torque).	
	0…1600%	Maximum torque limit.
20.07	MINIMUM TORQUE	FW block: LIMITS (see above).
	Defines the minimum torque limit for the drive (in percent of the motor nominal torque).	
	-1600…0%	Minimum torque limit.
20.08	THERM CURR LIM	FW block: None
	Enables the thermal current limitation. Thermal current limit is calculated by the inverter thermal protection function.	
	(0) Disable	The calculated thermal limit is not used. If the inverter output current is excessive, alarm IGBT OVERTEMP is generated and eventually the drive trips on fault IGBT OVERTEMP.
	(1) Enable	The calculated thermal current value limits the inverter output current (i.e. motor current).

Group 22 SPEED FEEDBACK

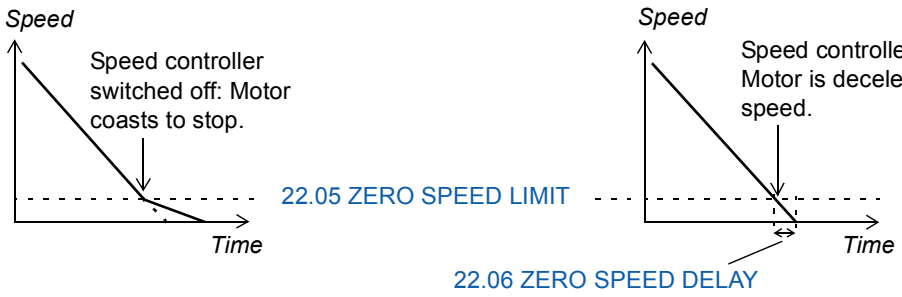
Settings for

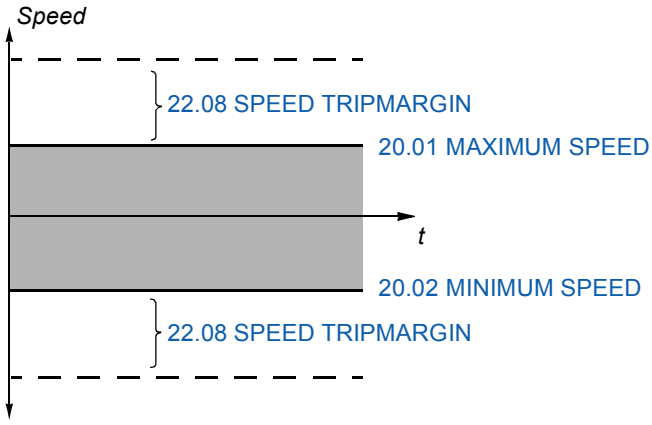
- selection of speed feedback used in drive control
- filtering disturbances in measured speed signal
- motor encoder gear function
- zero speed limit for stop function
- delay for Zero Speed Delay function
- definition of limits for actual speed supervision
- loss of speed feedback signal protection.



Firmware block: **SPEED FEEDBACK** (22)		
Block outputs located in other parameter groups		1.01 SPEED ACT (page 59)
22.01	SPEED FB SEL	FW block: SPEED FEEDBACK (see above)
	Selects the speed feedback value used in control.	
	(0) Estimated	Calculated speed estimate.
	(1) Enc1 speed	Actual speed measured with encoder 1. The encoder is selected by parameter 90.01 ENCODER 1 SEL.
	(2) Enc2 speed	Actual speed measured with encoder 2. The encoder is selected by parameter 90.02 ENCODER 2 SEL.
22.02	SPEED ACT FTIME	FW block: SPEED FEEDBACK (see above)
	Defines the time constant of the actual speed filter, i.e. time within the actual speed has reached 63% of the nominal speed (filtered speed = 1.01 SPEED ACT). If the used speed reference remains constant, the possible interferences in the speed measurement can be filtered with the actual speed filter. Reducing the ripple with filter may cause speed controller tuning problems. A long filter time constant and fast acceleration time contradict one another. A very long filter time results in unstable control. If there are substantial interferences in the speed measurement, the filter time constant should be proportional to the total inertia of the load and motor, in this case 10…30% of the mechanical time constant $t_{mech} = (n_{nom} / T_{nom}) \times J_{tot} \times 2\pi / 60$, where J_{tot} = total inertia of the load and motor (the gear ratio between the load and motor must be taken into account) n_{nom} = motor nominal speed T_{nom} = motor nominal torque See also parameter 26.06 SPD ERR FTIME.	
	0…10000 ms	Time constant for actual speed filter.

22.03	MOTOR GEAR MUL	FW block: SPEED FEEDBACK (see above)
	Defines the motor gear numerator for the motor encoder gear function. $\frac{\text{22.03 MOTOR GEAR MUL}}{\text{22.04 MOTOR GEAR DIV}} = \frac{\text{Actual speed}}{\text{Input speed}}$ where input speed is encoder 1/2 speed (1.08 ENCODER 1 SPEED / 1.10 ENCODER 2 SPEED) or speed estimate (1.14 SPEED ESTIMATED). **Note:** If the motor gear ratio differs from 1, the motor model uses an estimated speed instead of the speed feedback value. See also section *Motor encoder gear function* on page 48.	
	-2^{31}… 2^{31} -1	Numerator for motor encoder gear. **Note:** A setting of 0 is changed internally to 1.
22.04	MOTOR GEAR DIV	FW block: SPEED FEEDBACK (see above)
	Defines the motor gear denominator for the motor encoder gear function. See parameter 22.03 MOTOR GEAR MUL.	
	1 … 2^{31} -1	Denominator for motor encoder gear.
22.05	ZERO SPEED LIMIT	FW block: SPEED FEEDBACK (see above)
	Defines the zero speed limit. The motor is stopped along a speed ramp until the defined zero speed limit is reached. After the limit, the motor coasts to stop. **Note:** Too low a setting may result in the drive not stopping at all.	
	0…30000 rpm	Zero speed limit.

22.06	ZERO SPEED DELAY	FW block: SPEED FEEDBACK (see above)
	Defines the delay for the zero speed delay function. The function is useful in applications where a smooth and quick restarting is essential. During the delay the drive knows accurately the rotor position. **No Zero Speed Delay** **With Zero Speed Delay**  **No Zero Speed Delay** The drive receives a stop command and decelerates along a ramp. When the motor actual speed falls below the value of parameter 22.05 ZERO SPEED LIMIT, the speed controller is switched off. The inverter modulation is stopped and the motor coasts to standstill. **With Zero Speed Delay** The drive receives a stop command and decelerates along a ramp. When the actual motor speed falls below the value of parameter 22.05 ZERO SPEED LIMIT, the zero speed delay function activates. During the delay the function keeps the speed controller live: the inverter modulates, motor is magnetised and the drive is ready for a quick restart. Zero speed delay can be used e.g. with the jogging function.	
	0…30000 ms	Zero speed delay.
22.07	ABOVE SPEED LIM	FW block: SPEED FEEDBACK (see above)
	Defines the supervision limit for the actual speed. See also parameter 2.13 FBA MAIN SW, bit 10.	
	0…30000 rpm	Supervision limit for actual speed.

22.08	SPEED TRIPMARGIN	FW block: SPEED FEEDBACK (see above)
	Defines, together with 20.01 MAXIMUM SPEED and 20.02 MINIMUM SPEED, the maximum allowed speed of the motor (overspeed protection). If the actual speed (1.01 SPEED ACT) exceeds the speed limit defined by parameter 20.01 or 20.02 by more than 22.08 SPEED TRIPMARGIN, the drive trips on fault OVERSPEED. Example: If the maximum speed is 1420 rpm and speed trip margin is 300 rpm, the drive trips at 1720 rpm.  	
	0…10000 rpm	Speed trip margin.
22.09	SPEED FB FAULT	FW block: SPEED FEEDBACK (see above)
	Selects the action in case of speed feedback data loss. **Note:** If this parameter is set to (1) Warning or (2) No, a loss of feedback will cause an internal faulted state. To clear the internal fault and to reactivate speed feedback, use parameter 90.10 ENC PAR REFRESH.	
	(0) Fault	Drive trips on a fault (OPTION COMM LOSS, ENCODER 1/2, ENCODER 1/2 CABLE or SPEED FEEDBACK depending on the type of problem).
	(1) Warning	Drive continues operation with open loop control and generates an alarm (OPTION COMM LOSS, ENCODER 1/2 FAILURE, ENCODER 1/2 CABLE or SPEED FEEDBACK depending on the type of problem).
	(2) No	Drive continues operation with open loop control. No faults or alarms are generated.

Group 24 SPEED REF MOD

Settings for

- speed reference selection
- speed reference modification (scaling and inversion)
- constant speed and jogging references
- definition of absolute minimum speed reference.

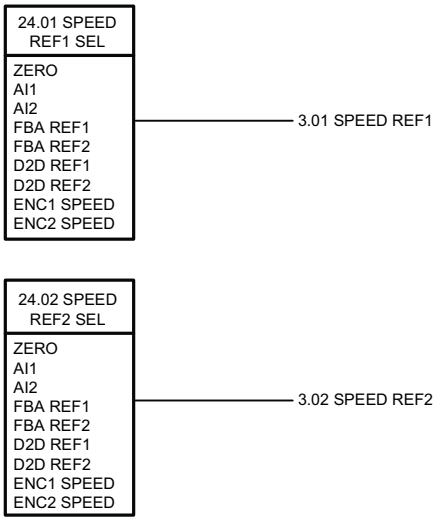
Depending on user selection, either speed reference 1 or speed reference 2 is active at a time.

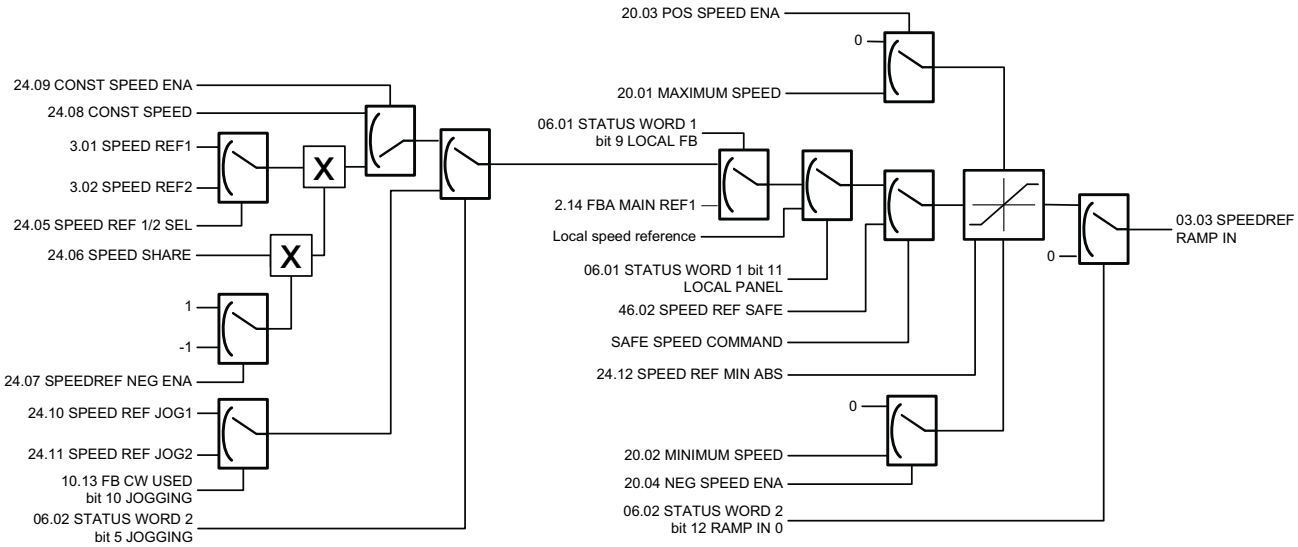
Speed reference can be any of the following (in priority order):

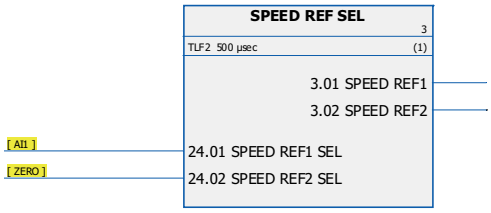
- fault speed reference (in a control panel or PC tool communication break)
- local speed reference (from panel)
- fieldbus local reference
- jogging reference 1/2
- constant speed reference 1/2
- external speed reference.

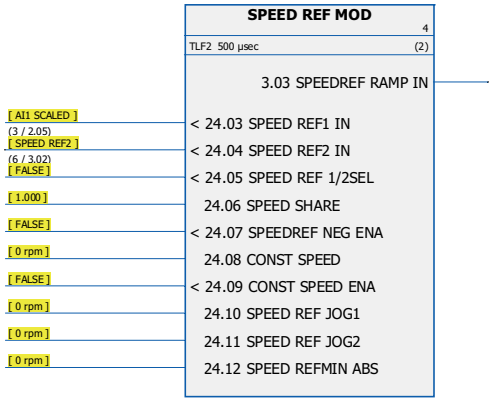
Note: Constant speed overrides external speed reference.

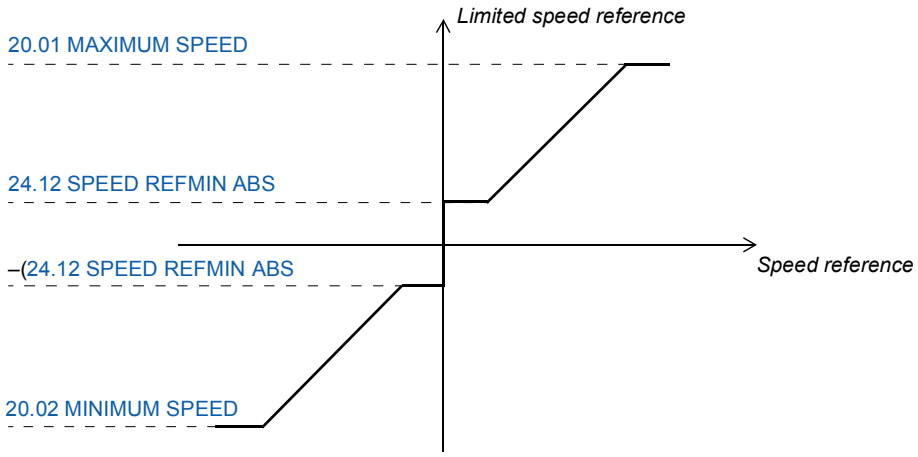
Speed reference is limited according to the set minimum and maximum speed values and ramped and shaped according to the defined acceleration and deceleration values. See parameter group 25 SPEED REF RAMP (page 114).





Firmware block: **SPEED REF SEL** (23) Selects the sources for two speed references, REF1 or REF2, from a selection list. Also shows the values of both speed references. The sources can alternatively be selected with value pointer parameters. See firmware block SPEED REF MOD on page 111.		
Block outputs located in other parameter groups		3.01 SPEED REF1 (page 68) 3.02 SPEED REF2 (page 68)
24.01	SPEED REF1 SEL	FW block: SPEED REF SEL (see above)
	Selects the source for speed reference 1 (3.01 SPEED REF1). Source for speed reference 1/2 can also be selected by value pointer parameter 24.03 SPEED REF1 IN / 24.04 SPEED REF2 IN.	
	(0) ZERO	Zero reference.
	(1) AI1	Analogue input AI1.
	(2) AI2	Analogue input AI2.
	(3) FBA REF1	Fieldbus reference 1.
	(4) FBA REF2	Fieldbus reference 2.
	(5) D2D REF1	Drive to drive reference 1.

	(6) D2D REF2	Drive to drive reference 2.
	(7) ENC1 SPEED	Encoder 1 (1.08 ENCODER 1 SPEED).
	(8) ENC2 SPEED	Encoder 2 (1.10 ENCODER 2 SPEED).
24.02	SPEED REF2 SEL	FW block: SPEED REF SEL (see above)
	Selects the source for speed reference 2 (3.02 SPEED REF2). See parameter 24.01 SPEED REF1 SEL.	
Firmware block: **SPEED REF MOD** (24) This block • selects the sources for two speed references, REF1 or REF2 • scales and inverts the speed reference • defines the constant speed reference • defines the speed reference for jogging functions 1 and 2 • defines the speed reference absolute minimum limit.		
Block outputs located in other parameter groups		3.03 SPEEDREF RAMP IN (page 68)
24.03	SPEED REF1 IN	FW block: SPEED REF MOD (see above)
	Selects the source for speed reference 1 (overrides the setting of parameter 24.01 SPEED REF1 SEL). The default value is P.3.1, i.e. 3.01 SPEED REF1, which is the output of the SPEED REF RAMP block.	
	Value pointer: Group and index	
24.04	SPEED REF2 IN	FW block: SPEED REF MOD (see above)
	Selects the source for speed reference 2 (overrides the setting of parameter 24.02 SPEED REF2 SEL). The default value is P.3.2, i.e. 3.02 SPEED REF2, which is the output of the SPEED REF RAMP block.	
	Value pointer: Group and index	
24.05	SPEED REF 1/2SEL	FW block: SPEED REF MOD (see above)
	Selects between speed reference 1 or 2. Reference 1/2 source is defined by parameter 24.03 SPEED REF1 IN / 24.04 SPEED REF2 IN. 0 = Speed reference 1.	
	Bit pointer: Group, index and bit	
24.06	SPEED SHARE	FW block: SPEED REF MOD (see above)
	Defines the scaling factor for speed reference 1/2 (speed reference 1 or 2 is multiplied by the defined value). Speed reference 1 or 2 is selected by parameter 24.05 SPEED REF 1/2SEL.	
	-8…8	Scaling factor for speed reference 1/2.

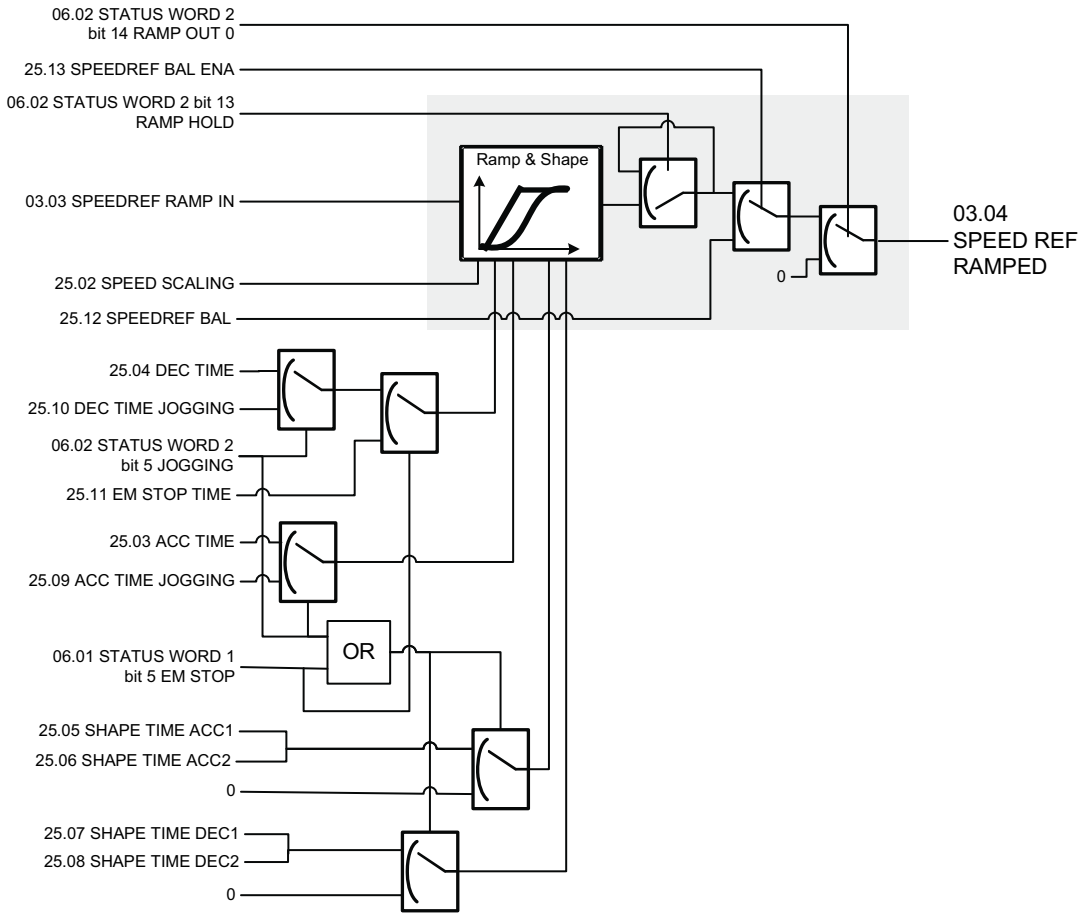
24.07	SPEEDREF NEG ENA	FW block: SPEED REF MOD (see above)
	Selects the source for the speed reference inversion. 1 = Sign of the speed reference is changed (inversion active).	
	Bit pointer: Group, index and bit	
24.08	CONST SPEED	FW block: SPEED REF MOD (see above)
	Defines the constant speed.	
	-30000…30000 rpm	Constant speed.
24.09	CONST SPEED ENA	FW block: SPEED REF MOD (see above)
	Selects the source for enabling the use of the constant speed reference define by parameter 24.08 CONST SPEED. 1 = Enable.	
	Bit pointer: Group, index and bit	
24.10	SPEED REF JOG1	FW block: SPEED REF MOD (see above)
	Defines the speed reference for jogging function 1. See section *Jogging* on page 45.	
	-30000…30000 rpm	Speed reference for jogging 1.
24.11	SPEED REF JOG2	FW block: SPEED REF MOD (see above)
	Defines the speed reference for jogging function 2. See section *Jogging* on page 45.	
	-30000…30000 rpm	Speed reference for jogging 2.
24.12	SPEED REFMIN ABS	FW block: SPEED REF MOD (see above)
	Defines the absolute minimum limit for the speed reference.  	
	0…30000 rpm	Absolute minimum limit for speed reference.

Group 25 SPEED REF RAMP

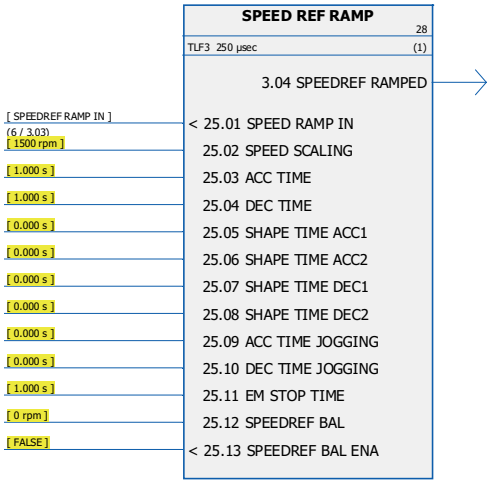
Speed reference ramp settings such as

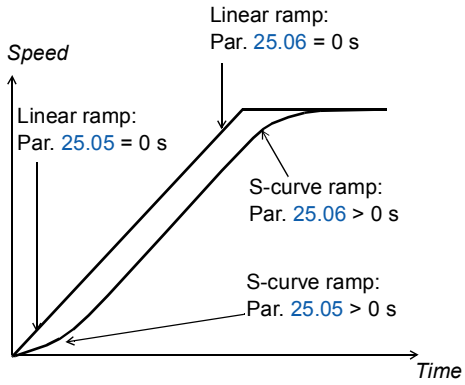
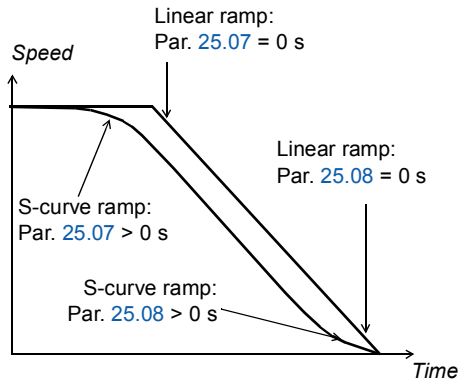
- selection of source for speed ramp input
- acceleration and deceleration times (also for jogging)
- acceleration and deceleration ramp shapes
- emergency stop OFF3 ramp time
- the speed reference balancing function (forcing the output of the ramp generator to a predefined value).

Note: Emergency stop OFF1 uses the currently active ramp time.



<table>
<tr><td colspan="2">Firmware block:
SPEED REF RAMP
(25)

This block
• selects the source for the speed ramp input
• adjusts acceleration and deceleration times (also for jogging)
• adjusts acceleration/deceleration ramp shapes
• adjusts ramp time for emergency stop OFF3
• forces the output of the ramp generator to a defined value
• shows the ramped and shaped speed reference value.</td><td> 
</td></tr>
<tr><td colspan="2">Block outputs located in other parameter groups</td><td>3.04 SPEEDREF RAMPED (page 68)</td></tr>
<tr><td>25.01</td><td>SPEED RAMP IN</td><td>FW block: SPEED REF RAMP (see above)</td></tr>
<tr><td></td><td colspan="2">Shows the source of the speed ramp input. The default value is P.3.3 i.e. signal 3.03 SPEEDREF RAMP IN, which is the output of the SPEED REF MOD firmware block.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
<tr><td>25.02</td><td>SPEED SCALING</td><td>FW block: SPEED REF RAMP (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the speed value used in acceleration and deceleration (parameters 25.03/25.09 and 25.04/25.10/25.11). Also affects fieldbus reference scaling (see Appendix A – Fieldbus control, section Fieldbus references on page 329).</td></tr>
<tr><td></td><td>0…30000 rpm</td><td>Speed value for acceleration/deceleration.</td></tr>
<tr><td>25.03</td><td>ACC TIME</td><td>FW block: SPEED REF RAMP (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the acceleration time i.e. the time required for the speed to change from zero to the speed value defined by parameter 25.02 SPEED SCALING.
If the speed reference increases faster than the set acceleration rate, the motor speed will follow the acceleration rate.
If the speed reference increases slower than the set acceleration rate, the motor speed will follow the reference signal.
If the acceleration time is set too short, the drive will automatically prolong the acceleration in order not to exceed the drive torque limits.</td></tr>
<tr><td></td><td>0…1800 s</td><td>Acceleration time.</td></tr>
</table>

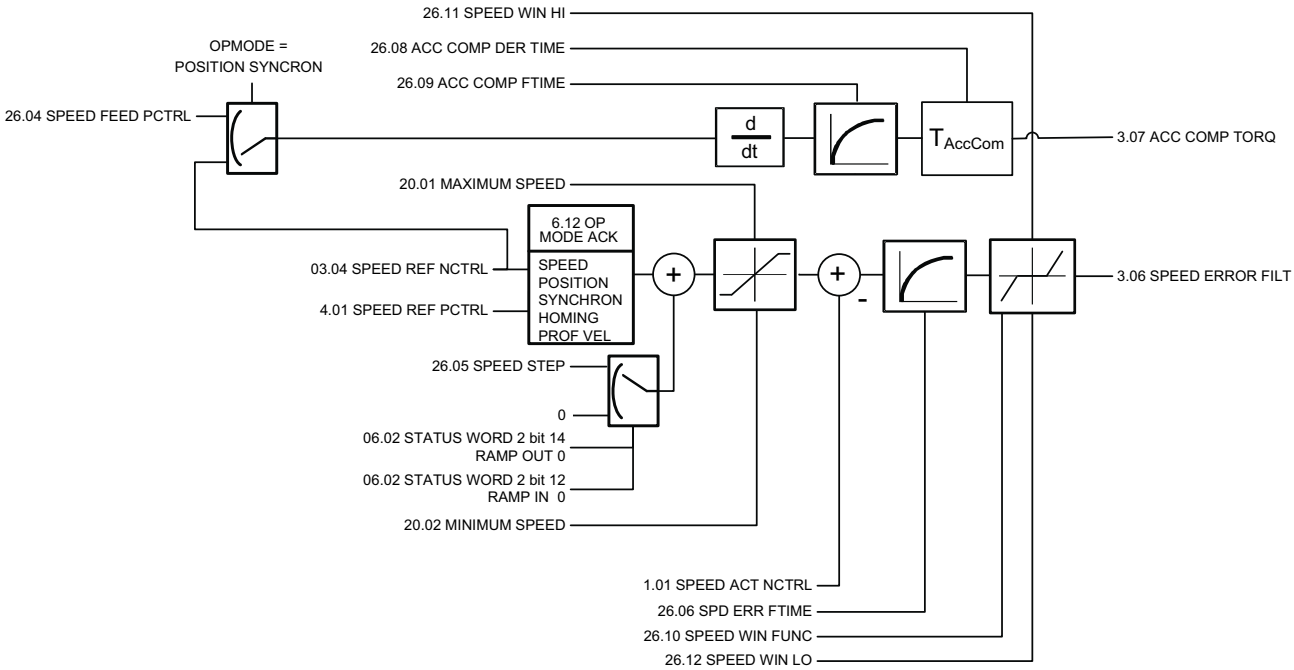
25.04	DEC TIME	FW block: SPEED REF RAMP (see above)
	Defines the deceleration time i.e. the time required for the speed to change from the speed value defined by parameter 25.02 SPEED SCALING to zero. If the speed reference decreases slower than the set deceleration rate, the motor speed will follow the reference signal. If the reference changes faster than the set deceleration rate, the motor speed will follow the deceleration rate. If the deceleration time is set too short, the drive will automatically prolong the deceleration in order not to exceed drive torque limits. If there is any doubt about the deceleration time being too short, ensure that the DC overvoltage control is on (parameter 47.01 OVERVOLTAGE CTRL). **Note:** If a short deceleration time is needed for a high inertia application, the drive should be equipped with an electric braking option e.g. with a brake chopper (built-in) and a brake resistor.	
	0…1800 s	Deceleration time.
25.05	SHAPE TIME ACC1	FW block: SPEED REF RAMP (see above)
	Selects the shape of the acceleration ramp at the beginning of the acceleration. 0.00 s: Linear ramp. Suitable for steady acceleration or deceleration and for slow ramps. 0.01…1000.00 s: S-curve ramp. S-curve ramps are ideal for conveyor and lifting applications. The S-curve consists of symmetrical curves at both ends of the ramp and a linear part in between. **Note:** When jogging or emergency ramp stop is active, acceleration and deceleration shape times are forced to zero.  	
	0…1000 s	Ramp shape at beginning of acceleration.
25.06	SHAPE TIME ACC2	FW block: SPEED REF RAMP (see above)
	Selects the shape of the acceleration ramp at the end of the acceleration. See parameter 25.05 SHAPE TIME ACC1.	
	0…1000 s	Ramp shape at end of acceleration.
25.07	SHAPE TIME DEC1	FW block: SPEED REF RAMP (see above)
	Selects the shape of the deceleration ramp at the beginning of the deceleration. See parameter 25.05 SHAPE TIME ACC1.	
	0…1000 s	Ramp shape at beginning of deceleration.

25.08	SHAPE TIME DEC2	FW block: SPEED REF RAMP (see above)
	Selects the shape of the deceleration ramp at the end of the deceleration. See parameter 25.05 SHAPE TIME ACC1.	
	0…1000 s	Ramp shape at end of deceleration.
25.09	ACC TIME JOGGING	FW block: SPEED REF RAMP (see above)
	Defines the acceleration time for the jogging function i.e. the time required for the speed to change from zero to the speed value defined by parameter 25.02 SPEED SCALING.	
	0…1800 s	Acceleration time for jogging.
25.10	DEC TIME JOGGING	FW block: SPEED REF RAMP (see above)
	Defines the deceleration time for the jogging function i.e. the time required for the speed to change from the speed value defined by parameter 25.02 SPEED SCALING to zero.	
	0…1800 s	Deceleration time for jogging.
25.11	EM STOP TIME	FW block: SPEED REF RAMP (see above)
	Defines the time inside which the drive is stopped if an emergency stop OFF3 is activated (i.e. the time required for the speed to change from the speed value defined by parameter 25.02 SPEED SCALING to zero). Emergency stop activation source is selected by parameter 10.10 EM STOP OFF3. Emergency stop can also be activated through fieldbus (2.12 FBA MAIN CW). Emergency stop OFF1 uses the active ramp time.	
	0…1800 s	Emergency stop OFF3 deceleration time.
25.12	SPEEDREF BAL	FW block: SPEED REF RAMP (see above)
	Defines the reference for the speed ramp balancing, i.e. the output of the speed reference ramp firmware block is forced to a defined value. The source for the balancing enable signal is selected by parameter 25.13 SPEEDREF BAL.	
	-30000…30000 rpm	Speed ramp balancing reference.
25.13	SPEEDREF BAL ENA	FW block: SPEED REF RAMP (see above)
	Selects the source for enabling the speed ramp balancing. See parameter 25.12 SPEEDREF BAL. 1 = Speed ramp balancing enabled.	
	Bit pointer: Group, index and bit	

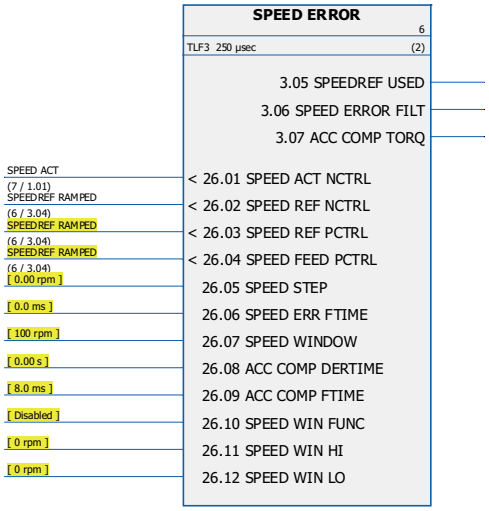
Group 26 SPEED ERROR

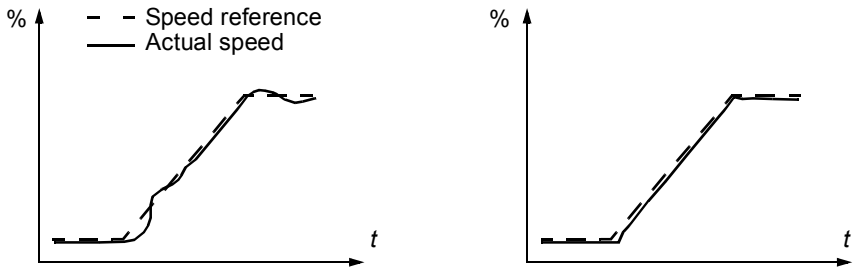
Speed error is determined by comparing the speed reference and speed feedback. The error can be filtered using a first-order low-pass filter if the feedback and reference have disturbances. In addition, a torque boost can be applied to compensate acceleration; the torque is relative to the rate of change (derivative) in the speed reference and inertia of the load. The speed error value can be supervised using the window function.

- The signal used as speed reference is 3.04 SPEEDREF RAMPED.



<table>
<tr><td colspan="2">Firmware block:
SPEED ERROR
(26)

This block
• selects the source for speed error calculation (speed reference - actual speed) in different control modes
• selects the source for speed reference
• defines the speed error filtering time
• defines an additional speed step to the speed error
• defines the supervision of speed error with the speed error window function
• defines inertia compensation during acceleration
• shows the used speed reference, filtered speed error and the output of the acceleration compensation.</td><td> 
</td></tr>
<tr><td colspan="2">Block outputs located in other parameter groups</td><td>3.05 SPEEDREF USED (page 68)
3.06 SPEED ERROR FILT (page 68)
3.07 ACC COMP TORQ (page 68)</td></tr>
<tr><td>26.01</td><td>SPEED ACT NCTRL</td><td>FW block: SPEED ERROR (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source for the actual speed in the speed control mode.
Note: This parameter has been locked, i.e. no user setting is possible.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
<tr><td>26.02</td><td>SPEED REF NCTRL</td><td>FW block: SPEED ERROR (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source for the speed reference in the speed control mode.
Note: This parameter has been locked, i.e. no user setting is possible.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
<tr><td>26.03</td><td>SPEED REF PCTRL</td><td>FW block: SPEED ERROR (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source for the speed reference in position and synchron control modes.
Note: This parameter is only for positioning applications.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
<tr><td>26.04</td><td>SPEED FEED PCTRL</td><td>FW block: SPEED ERROR (see above)</td></tr>
<tr><td></td><td colspan="2">This parameter is only for positioning applications.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
</table>

26.05	SPEED STEP	FW block: SPEED ERROR (see above)
	Defines an additional speed step given to the input of the speed controller (added to the speed error value).	
	-30000…30000 rpm	Speed step.
26.06	SPD ERR FTIME	FW block: SPEED ERROR (see above)
	Defines the time constant of the speed error low pass filter. If the used speed reference changes rapidly (servo application), the possible interferences in the speed measurement can be filtered with the speed error filter. Reducing the ripple with filter may cause speed controller tuning problems. A long filter time constant and fast acceleration time contradict one another. A very long filter time results in unstable control. See also parameter 22.02 SPEED ACT FTIME.	
	0…1000 ms	Time constant for speed error low pass filter. 0 ms = filtering disabled.
26.07	SPEED WINDOW	FW block: SPEED ERROR (see above)
	Defines the absolute value for the motor speed window supervision, i.e. the absolute value for the difference between the actual speed and the unramped speed reference (1.01 SPEED ACT - 3.03 SPEEDREF RAMP IN). When the motor speed is within the limits defined by this parameter, signal 2.13 bit 8 (AT_SETPOINT) value is 1. If the motor speed is not within the defined limits, bit 8 value is 0.	
	0…30000 rpm	Absolute value for motor speed window supervision.
26.08	ACC COMP DERTIME	FW block: SPEED ERROR (see above)
	Defines the derivation time for acceleration (deceleration) compensation. Used to improve the speed control dynamic reference change. In order to compensate inertia during acceleration, a derivative of the speed reference is added to the output of the speed controller. The principle of a derivative action is described for parameter 28.04 DERIVATION TIME. **Note:** The parameter value should be proportional to the total inertia of the load and motor, i.e. approximately 50…100% of the mechanical time constant (t_{mech}). See the mechanical time constant equation in parameter 22.02 SPEED ACT FTIME. If parameter value is set to zero, the function is deactivated. The figure below shows the speed responses when a high inertia load is accelerated along a ramp. **No acceleration compensation** **With acceleration compensation**  See also parameter 26.09 ACC COMP FTIME. The source for the acceleration compensation torque can also be selected by parameter 28.06 ACC COMPENSATION. See firmware group 28 SPEED CONTROL.	

	0…600 s	Derivation time for acceleration/deceleration compensation.
26.09	ACC COMP FTIME	FW block: SPEED ERROR (see above)
	Defines the filter time for the acceleration compensation.	
	0…1000 ms	Filter time for acceleration compensation. 0 ms = filtering disabled.
26.10	SPEED WIN FUNC	FW block: SPEED ERROR (see above)
	Enables or disables speed error window control. Speed error window control forms a speed supervision function for a torque-controlled drive. It supervises the speed error value (speed reference – actual speed). In the normal operating range, window control keeps the speed controller input at zero. When the speed error moves outside the window, the exceeding part of the error value is connected to the speed controller. The speed controller produces a reference term relative to the input and gain of the speed controller (parameter 28.02 PROPORT GAIN) which the torque selector adds (if selected in parameter group 34) to the torque reference. The result is used as the internal torque reference for the drive (3.13 TORQ REF TO TC). Example: In a load loss condition, the internal torque reference of the drive (3.13 TORQ REF TO TC) is decreased to prevent an excessive rise of the motor speed. If window control were inactive, the motor speed would rise until a speed limit of the drive were reached.	
	(0) Disabled	Speed error window control inactive.
	(1) Absolute	Speed error window control active. The window boundaries set by parameters 26.11 and 26.12 are effective in both directions of rotation (the boundaries have a negative value when actual speed is negative).
	(2) Relative	Speed error window control active. The window boundaries set by parameters 26.11 and 26.12 are only effective in the forward direction (i.e. when actual speed is positive).
26.11	SPEED WIN HI	FW block: SPEED ERROR (see above)
	High limit for speed window control. See parameter 26.10 SPEED WIN FUNC.	
	0…3000 rpm	High limit for speed error window control.
26.12	SPEED WIN LO	FW block: SPEED ERROR (see above)
	Low limit for speed window control. See parameter 26.10 SPEED WIN FUNC.	
	0…3000 rpm	Low limit for speed error window control.

Group 28 SPEED CONTROL

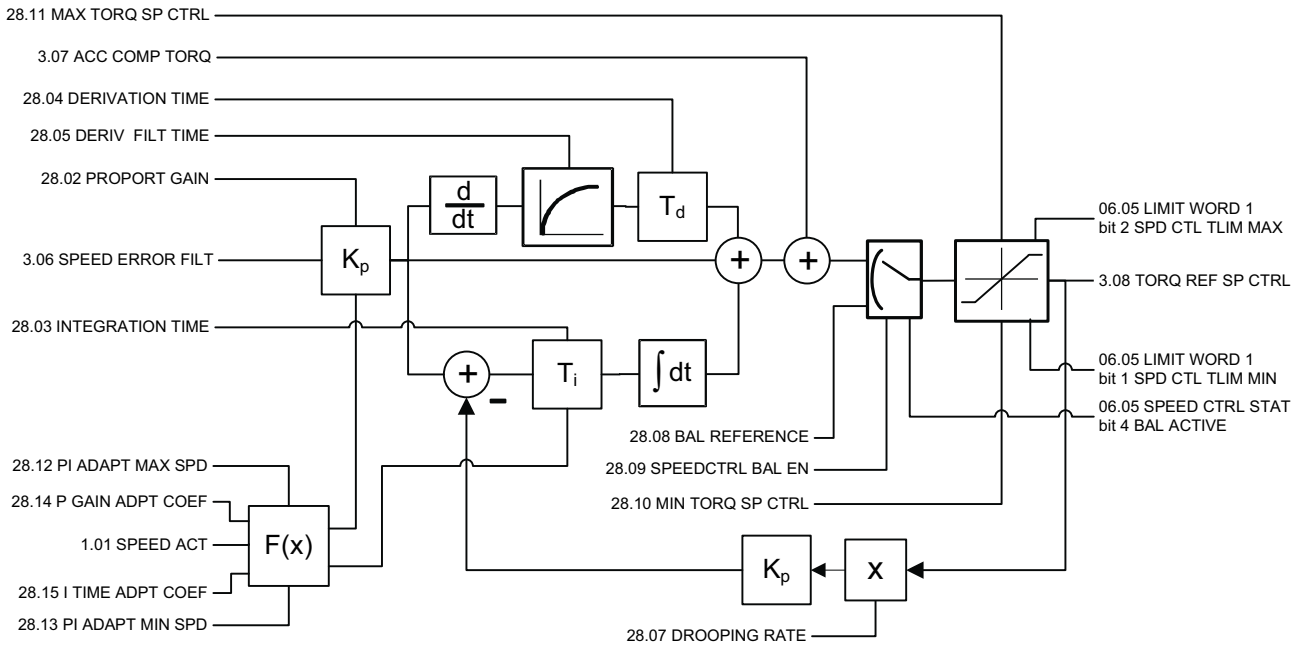
Speed controller settings such as

- selection of source for speed error
- adjustment of PID-type speed controller variables
- limitation of speed controller output torque
- selection of source for acceleration compensation torque
- forcing an external value to the output of the speed controller (with the balancing function).
- adjustment of the load sharing in a Master/Follower application run by several drives (the drooping function).

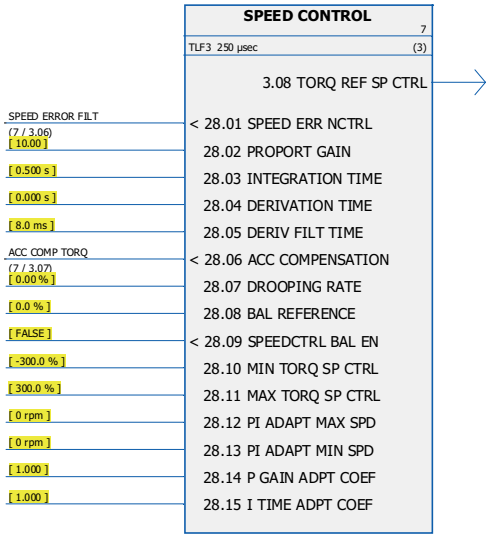
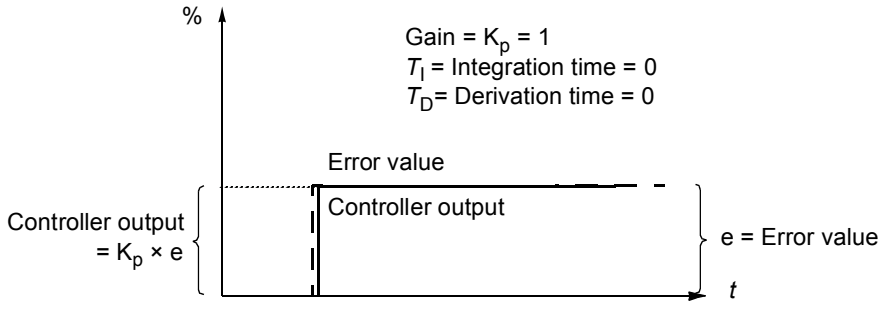
The speed controller includes an anti-windup function (i.e. I-term is frozen during torque reference limitation).

In torque control mode, the speed controller output is frozen.

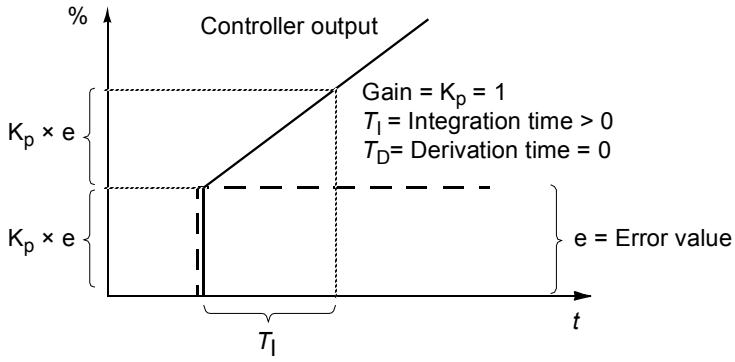
For manual speed controller tuning, see section *Speed controller tuning* on page 26.

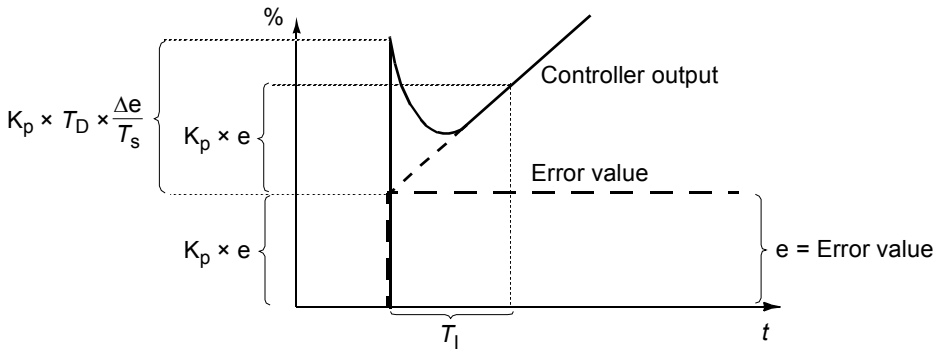


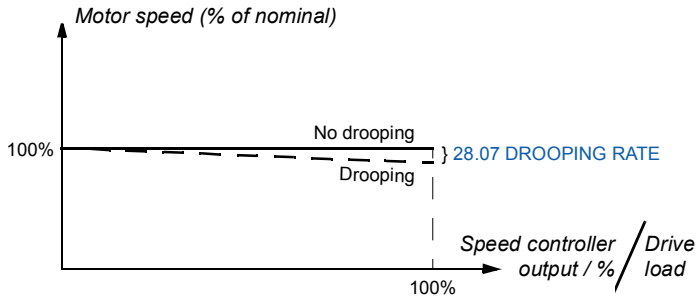
<table>
<tr><td>Firmware block:
SPEED CONTROL
(28)

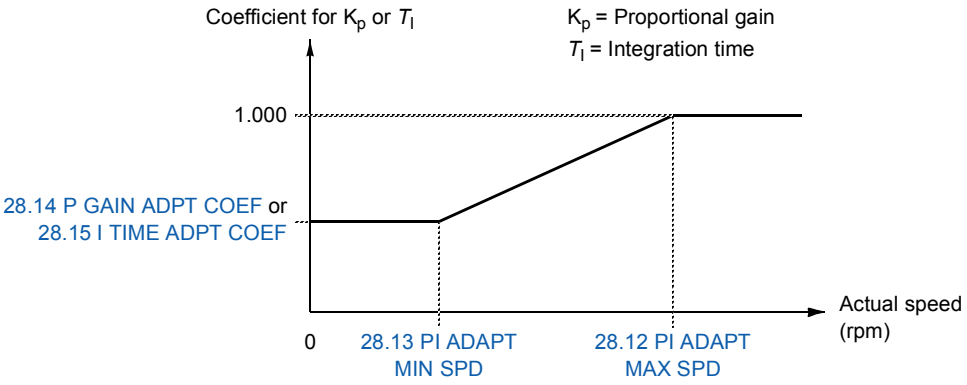
This block
• selects the source for speed error
• adjusts PID-type speed controller variables
• defines limits for speed controller output torque
• selects the source for acceleration compensation torque
• configures the balancing function which forces the output of the speed controller to an external value
• configures the drooping function (adjustment of load sharing in a Master/Follower application)
• shows the limited speed controller output torque value.</td><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Block outputs located in other parameter groups</td><td>3.08 TORQ REF SP CTRL (page 68)</td></tr>
<tr><td>28.01</td><td>SPEED ERR NCTRL</td><td>FW block: SPEED CONTROL (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source for the speed error (reference - actual). The default value is P.3.6 i.e. parameter 3.06 SPEED ERROR FILT, which is the output of the SPEED ERROR firmware block.
Note: This parameter has been locked, i.e. no user setting is possible.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
<tr><td>28.02</td><td>PROPORT GAIN</td><td>FW block: SPEED CONTROL (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the proportional gain (K_p) of the speed controller. Too large a gain may cause speed oscillation. The figure below shows the speed controller output after an error step when the error remains constant.
 

If gain is set to 1, a 10% change in error value (reference - actual value) causes the speed controller output to change by 10%.
Note: This parameter is automatically set by the speed controller autotune function. See parameter 28.16 PI TUNE MODE.</td></tr>
<tr><td></td><td>0…200</td><td>Proportional gain for speed controller.</td></tr>
</table>

28.03	INTEGRATION TIME	FW block: SPEED CONTROL (see above)
	Defines the integration time of the speed controller. The integration time defines the rate at which the controller output changes when the error value is constant and the proportional gain of the speed controller is 1. The shorter the integration time, the faster the continuous error value is corrected. Too short integration time makes the control unstable. If parameter value is set to zero, the I-part of the controller is disabled. Anti-windup stops the integrator if the controller output is limited. See 6.05 LIMIT WORD 1. The figure below shows the speed controller output after an error step when the error remains constant.  **Note:** This parameter is automatically set by the speed controller autotune function. See parameter 28.16 PI TUNE MODE.	
	0…600 s	Integration time for speed controller.

28.04	DERIVATION TIME	FW block: SPEED CONTROL (see above)
	Defines the derivation time of the speed controller. Derivative action boosts the controller output if the error value changes. The longer the derivation time, the more the speed controller output is boosted during the change. If the derivation time is set to zero, the controller works as a PI controller, otherwise as a PID controller. The derivation makes the control more responsive for disturbances. The speed error derivative must be filtered with a low pass filter to eliminate disturbances. The figure below shows the speed controller output after an error step when the error remains constant. Gain = K_p = 1 T_I = Integration time > 0 T_D= Derivation time > 0 T_s= Sample time period = 250 µs e = Error value ∆e = Error value change between two samples  **Note:** Changing this parameter value is recommended only if a pulse encoder is used.	
	0…10 s	Derivation time for speed controller.
28.05	DERIV FILT TIME	FW block: SPEED CONTROL (see above)
	Defines the derivation filter time constant.	
	0…1000 ms	Derivation filter time constant.
28.06	ACC COMPENSATION	FW block: SPEED CONTROL (see above)
	Selects the source for the acceleration compensation torque. The default value is P.3.7, i.e signal 3.07 ACC COMP TORQ, which is the output of the SPEED ERROR firmware block. **Note:** This parameter has been locked, i.e. no user setting is possible.	
	Value pointer: Group and index	

28.07	DROOPING RATE	FW block: SPEED CONTROL (see above)
	Defines the droop rate (in percent of the motor nominal speed). The drooping slightly decreases the drive speed as the drive load increases. The actual speed decrease at a certain operating point depends on the droop rate setting and the drive load (= torque reference / speed controller output). At 100% speed controller output, drooping is at its nominal level, i.e. equal to the value of this parameter. The drooping effect decreases linearly to zero along with the decreasing load. Drooping rate can be used e.g. to adjust the load sharing in a Master/Follower application run by several drives. In a Master/Follower application the motor shafts are coupled to each other. The correct droop rate for a process must be found out case by case in practice. **Speed decrease =** Speed controller output × Drooping × Max. speed **Example:** Speed controller output is 50%, drooping rate is 1%, maximum speed of the drive is 1500 rpm. Speed decrease = 0.50 × 0.01 × 1500 rpm = 7.5 rpm.  	
	0…100%	Droop rate.
28.08	BAL REFERENCE	FW block: SPEED CONTROL (see above)
	Defines the reference used in the speed control output balancing, i.e. an external value to be forced to the output of the speed controller. In order to guarantee smooth operation during output balancing, the speed controller D-part is disabled and the acceleration compensation term is set to zero. The source for the balancing enable signal is selected by parameter 28.09 SPEEDCTRL BAL EN.	
	-1600…1600%	Speed control output balancing reference.
28.09	SPEEDCTRL BAL EN	FW block: SPEED CONTROL (see above)
	Selects the source for the speed control output balancing enable signal. See parameter 28.08 BAL REFERENCE. 1 = Enabled. 0 = Disabled.	
	Bit pointer: Group, index and bit	
28.10	MIN TORQ SP CTRL	FW block: SPEED CONTROL (see above)
	Defines the minimum speed controller output torque.	
	-1600…1600%	Minimum speed controller output torque.
28.11	MAX TORQ SP CTRL	FW block: SPEED CONTROL (see above)
	Defines the maximum speed controller output torque.	
	-1600…1600%	Maximum speed controller output torque.

28.12	PI ADAPT MAX SPD	FW block: SPEED CONTROL (see above)
	Maximum actual speed for speed controller adaptation. Speed controller gain and integration time can be adapted according to actual speed. This is done by multiplying the gain (28.02 PROPORT GAIN) and integration time (28.03 INTEGRATION TIME) by coefficients at certain speeds. The coefficients are defined individually for both gain and integration time. When the actual speed is below or equal to 28.13 PI ADAPT MIN SPD, 28.02 PROPORT GAIN and 28.03 INTEGRATION TIME are multiplied by 28.14 P GAIN ADPT COEF and 28.15 I TIME ADPT COEF respectively. When the actual speed is equal to or exceeds 28.12 PI ADAPT MAX SPD, no adaptation takes place; in other words, 28.02 PROPORT GAIN and 28.03 INTEGRATION TIME are used as such. Between 28.13 PI ADAPT MIN SPD and 28.12 PI ADAPT MAX SPD, the coefficients are calculated linearly on the basis of the breakpoints. 	
	0…30000 rpm	Maximum actual speed for speed controller adaptation.
28.13	PI ADAPT MIN SPD	FW block: SPEED CONTROL (see above)
	Minimum actual speed for speed controller adaptation. See parameter 28.12 PI ADAPT MAX SPD.	
	0…30000 rpm	Minimum actual speed for speed controller adaptation.
28.14	P GAIN ADPT COEF	FW block: SPEED CONTROL (see above)
	Proportional gain coefficient. See parameter 28.12 PI ADAPT MAX SPD.	
	0.000 … 10.000	Proportional gain coefficient.
28.15	I TIME ADPT COEF	FW block: SPEED CONTROL (see above)
	Integration time coefficient. See parameter 28.12 PI ADAPT MAX SPD.	
	0.000 … 10.000	Integration time coefficient.

28.16	PI TUNE MODE	FW block: None

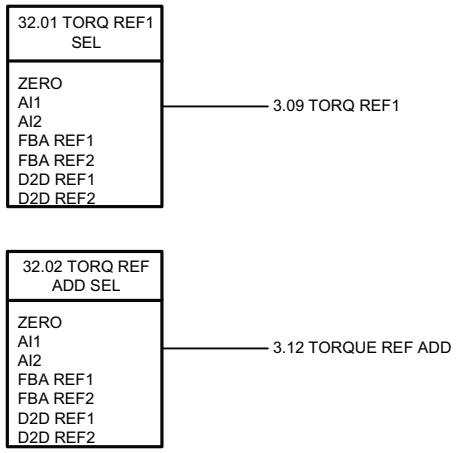
	Activates the speed controller autotune function.
	The autotune will automatically set parameters 28.02 PROPORT GAIN and 28.03 INTEGRATION TIME, as well as 1.31 MECH TIME CONST. If the User autotune mode is chosen, also 26.06 SPD ERR FTIME is automatically set.
	The status of the autotune routine is shown by parameter 6.03 SPEED CTRL STAT.
	WARNING! The motor will reach the torque and current limits during the autotune routine. ENSURE THAT IT IS SAFE TO RUN THE MOTOR BEFORE PERFORMING THE AUTOTUNE ROUTINE!
	Notes:
	• Before using the autotune function, the following parameters should be set:
	  • All parameters adjusted during the start-up as described in chapter *Start-up* (page 15)
	  • 22.05 ZERO SPEED LIMIT
	  • Speed scaling and reference ramp settings in Group 25 SPEED REF RAMP
	  • 26.06 SPD ERR FTIME
	  • If the User autotune mode is desired: 28.17 TUNE BANDWIDTH and 28.18 TUNE DAMPING.
	• The drive must be in local control mode and stopped before an autotune is requested.
	• After requesting an autotune with this parameter, start the drive within 20 seconds.
	• Wait until the autotune routine is completed (this parameter has reverted to the value (0) Done). The routine can be aborted by stopping the drive.
	• Check the values of the parameters set by the autotune function.
	See also section *Speed controller tuning* on page 46.	

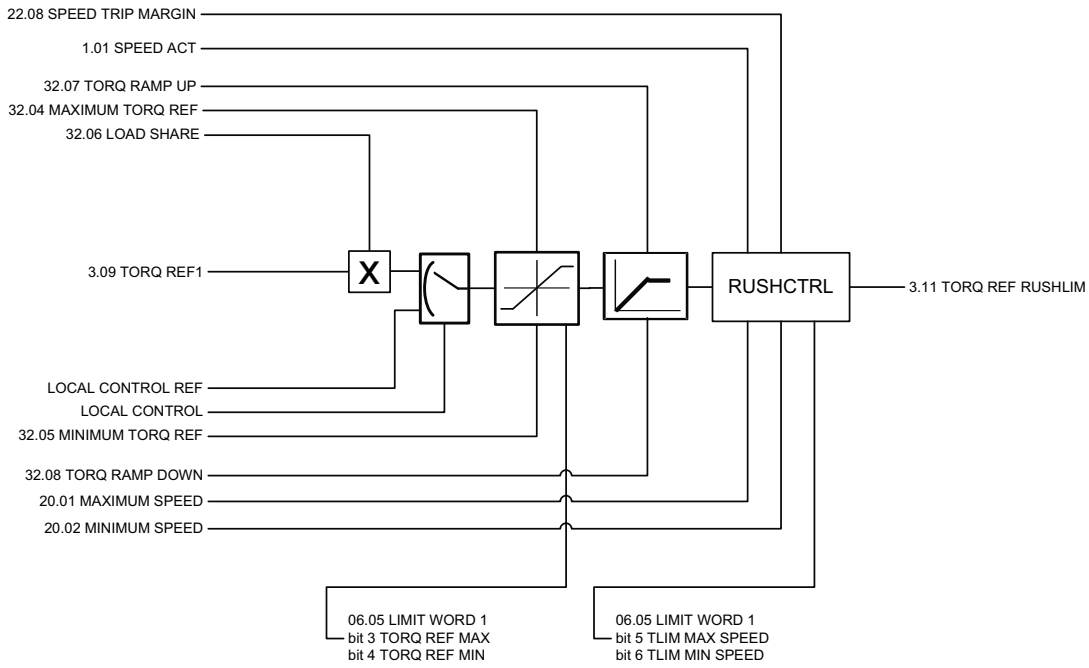
	(0) Done	No tuning has been requested (normal operation). The parameter also reverts to this value after an autotune is completed.
	(1) Smooth	Request speed controller autotune with preset settings for smooth operation.
	(2) Middle	Request speed controller autotune with preset settings for medium-tight operation.
	(3) Tight	Request speed controller autotune with preset settings for tight operation.
	(4) User	Request speed controller autotune with the settings defined by parameters 28.17 TUNE BANDWIDTH and 28.18 TUNE DAMPING.
28.17	TUNE BANDWIDTH	FW block: None
	Speed controller bandwidth for autotune procedure, User mode (see parameter 28.16 PI TUNE MODE). A larger bandwidth results in more restricted speed controller settings.	
	0.00 … 2000.00 Hz	Tune bandwidth for User autotune mode.
28.18	TUNE DAMPING	FW block: None
	Speed controller damping for autotune procedure, User mode (see parameter 28.16 PI TUNE MODE). Higher damping results in safer and smoother operation.	
	0.0 … 200.0	Speed controller damping for User autotune mode.

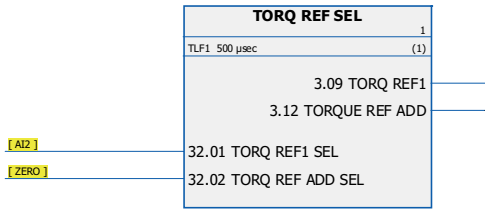
Group 32 TORQUE REFERENCE

Reference settings for torque control.

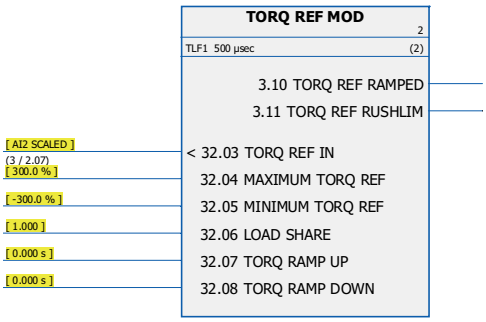
In torque control, the drive speed is limited between the defined minimum and maximum limits. Speed-related torque limits are calculated and the input torque reference is limited according to these limits. An OVERSPEED fault is generated if the maximum allowed speed is exceeded.





Firmware block: **TORQ REF SEL** (32) Selects the source for torque reference 1 (from a parameter selection list) and the source for torque reference addition (used e.g. for compensating mechanical interferences). Also shows the torque reference and reference addition values.		
Block outputs located in other parameter groups		3.09 TORQ REF1 (page 68) 3.12 TORQUE REF ADD (page 68)
32.01	TORQ REF1 SEL	FW block: TORQ REF SEL (see above)
	Selects the source for torque reference 1. See also parameter 32.03 TORQ REF IN.	
	(0) ZERO	Zero reference.
	(1) AI1	Analogue input AI1.
	(2) AI2	Analogue input AI2.
	(3) FBA REF1	Fieldbus reference 1.
	(4) FBA REF2	Fieldbus reference 2.
	(5) D2D REF1	Drive to drive reference 1.
	(6) D2D REF2	Drive to drive reference 2.
32.02	TORQ REF ADD SEL	FW block: TORQ REF SEL (see above)
	Selects the source for the torque reference addition, 3.12 TORQUE REF ADD. Parameter 34.10 TORQ REF ADD SRC is connected to signal 3.12 TORQUE REF ADD by default. Because the reference is added after the torque reference selection, this parameter can be used in speed and torque control modes. See block diagram at parameter group 34 REFERENCE CTRL (page 136).	
	(0) ZERO	Zero reference addition.
	(1) AI1	Analogue input AI1.
	(2) AI2	Analogue input AI2.
	(3) FBA REF1	Fieldbus reference 1.
	(4) FBA REF2	Fieldbus reference 2.
	(5) D2D REF1	Drive to drive reference 1.
	(6) D2D REF2	Drive to drive reference 2.

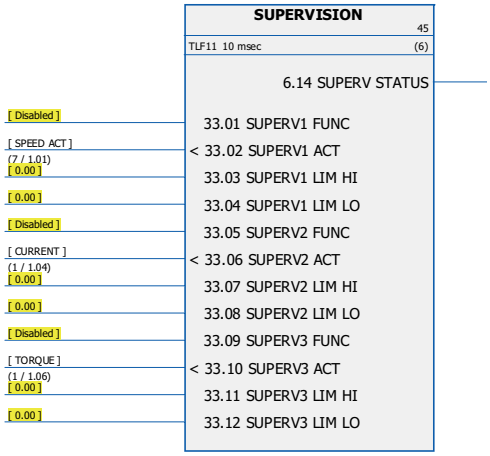
<table>
<tr><td colspan="2">Firmware block:
TORQ REF MOD
(33)

This block
• selects the source for the torque reference
• scales the input torque reference according to the defined load share factor
• defines limits for the torque reference
• defines ramp-up and ramp-down times for the torque reference
• shows the ramped torque reference value and the torque reference value limited by the rush control.</td><td> 
</td></tr>
<tr><td colspan="2">Block outputs located in other parameter groups</td><td>3.10 TORQ REF RAMPED (page 68)
3.11 TORQ REF RUSHLIM (page 68)</td></tr>
<tr><td>32.03</td><td>TORQ REF IN</td><td>FW block: TORQ REF MOD (see above)</td></tr>
<tr><td></td><td colspan="2">Selects the source for the torque reference input for the torque ramp function. The default value is P.3.9, i.e. signal 3.09 TORQ REF1, which is the output of the TORQ REF SEL firmware block.</td></tr>
<tr><td></td><td colspan="2">Value pointer: Group and index</td></tr>
<tr><td>32.04</td><td>MAXIMUM TORQ REF</td><td>FW block: TORQ REF MOD (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the maximum torque reference.</td></tr>
<tr><td></td><td>0…1000%</td><td>Maximum torque reference.</td></tr>
<tr><td>32.05</td><td>MINIMUM TORQ REF</td><td>FW block: TORQ REF MOD (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the minimum torque reference.</td></tr>
<tr><td></td><td>-1000…0%</td><td>Minimum torque reference.</td></tr>
<tr><td>32.06</td><td>LOAD SHARE</td><td>FW block: TORQ REF MOD (see above)</td></tr>
<tr><td></td><td colspan="2">Scales the external torque reference to a required level (external torque reference is multiplied by the selected value).
Note: If local torque reference is used, no load share scaling is applied.</td></tr>
<tr><td></td><td>-8…8</td><td>External torque reference multiplier.</td></tr>
<tr><td>32.07</td><td>TORQ RAMP UP</td><td>FW block: TORQ REF MOD (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the torque reference ramp-up time, i.e. the time for the reference to increase from zero to the nominal motor torque.</td></tr>
<tr><td></td><td>0…60 s</td><td>Torque reference ramp-up time.</td></tr>
<tr><td>32.08</td><td>TORQ RAMP DOWN</td><td>FW block: TORQ REF MOD (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the torque reference ramp-down time, i.e. the time for the reference to decrease from the nominal motor torque to zero.</td></tr>
</table>

	0…60 s	Torque reference ramp-down time.

Group 33 SUPERVISION

Configuration of signal supervision.

Firmware block: **SUPERVISION** (17)		
Block outputs located in other parameter groups		6.14 SUPERV STATUS (page 73)
33.01	SUPERV1 FUNC	FW block: SUPERVISION (see above)
	Selects the mode of supervision 1.	
	(0) Disabled	Supervision 1 not in use.
	(1) Low	When the signal selected by parameter 33.02 SUPERV1 ACT falls below the value of parameter 33.04 SUPERV1 LIM LO, bit 0 of 6.14 SUPERV STATUS is activated. To clear the bit, the signal must exceed the value of parameter 33.03 SUPERV1 LIM HI.
	(2) High	When the signal selected by parameter 33.02 SUPERV1 ACT exceeds the value of parameter 33.03 SUPERV1 LIM HI, bit 0 of 6.14 SUPERV STATUS is activated. To clear the bit, the signal must fall below the value of parameter 33.04 SUPERV1 LIM LO.
	(3) Abs Low	When the absolute value of the signal selected by parameter 33.02 SUPERV1 ACT falls below the value of parameter 33.04 SUPERV1 LIM LO, bit 0 of 6.14 SUPERV STATUS is activated. To clear the bit, the absolute value of the signal must exceed the value of parameter 33.03 SUPERV1 LIM HI.
	(4) Abs High	When the absolute value of the signal selected by parameter 33.02 SUPERV1 ACT exceeds the value of parameter 33.03 SUPERV1 LIM HI, bit 0 of 6.14 SUPERV STATUS is activated. To clear the bit, the absolute value of the signal must fall below the value of parameter 33.04 SUPERV1 LIM LO.
33.02	SUPERV1 ACT	FW block: SUPERVISION (see above)
	Selects the signal to be monitored by supervision 1. See parameter 33.01 SUPERV1 FUNC.	
	Value pointer: Group and index	

33.03	SUPERV1 LIM HI	FW block: SUPERVISION (see above)
	Sets the upper limit for supervision 1. See parameter 33.01 SUPERV1 FUNC.	
	-32768…32768	Upper limit for supervision 1.
33.04	SUPERV1 LIM LO	FW block: SUPERVISION (see above)
	Sets the lower limit for supervision 1. See parameter 33.01 SUPERV1 FUNC.	
	-32768…32768	Lower limit for supervision 1.
33.05	SUPERV2 FUNC	FW block: SUPERVISION (see above)
	Selects the mode of supervision 2.	
	(0) Disabled	Supervision 2 not in use.
	(1) Low	When the signal selected by parameter 33.06 SUPERV2 ACT falls below the value of parameter 33.08 SUPERV2 LIM LO, bit 1 of 6.14 SUPERV STATUS is activated. To clear the bit, the signal must exceed the value of parameter 33.07 SUPERV2 LIM HI.
	(2) High	When the signal selected by parameter 33.06 SUPERV2 ACT exceeds the value of parameter 33.07 SUPERV2 LIM HI, bit 1 of 6.14 SUPERV STATUS is activated. To clear the bit, the signal must fall below the value of parameter 33.08 SUPERV2 LIM LO.
	(3) Abs Low	When the absolute value of the signal selected by parameter 33.06 SUPERV2 ACT falls below the value of parameter 33.08 SUPERV2 LIM LO, bit 1 of 6.14 SUPERV STATUS is activated. To clear the bit, the absolute value of the signal must exceed the value of parameter 33.07 SUPERV2 LIM HI.
	(4) Abs High	When the absolute value of the signal selected by parameter 33.06 SUPERV2 ACT exceeds the value of parameter 33.07 SUPERV2 LIM HI, bit 1 of 6.14 SUPERV STATUS is activated. To clear the bit, the absolute value of the signal must fall below the value of parameter 33.08 SUPERV2 LIM LO.
33.06	SUPERV2 ACT	FW block: SUPERVISION (see above)
	Selects the signal to be monitored by supervision 2. See parameter 33.05 SUPERV2 FUNC.	
	Value pointer: Group and index	
33.07	SUPERV2 LIM HI	FW block: SUPERVISION (see above)
	Sets the upper limit for supervision 2. See parameter 33.05 SUPERV2 FUNC.	
	-32768…32768	Upper limit for supervision 2.
33.08	SUPERV2 LIM LO	FW block: SUPERVISION (see above)
	Sets the lower limit for supervision 2. See parameter 33.05 SUPERV2 FUNC.	
	-32768…32768	Lower limit for supervision 2.

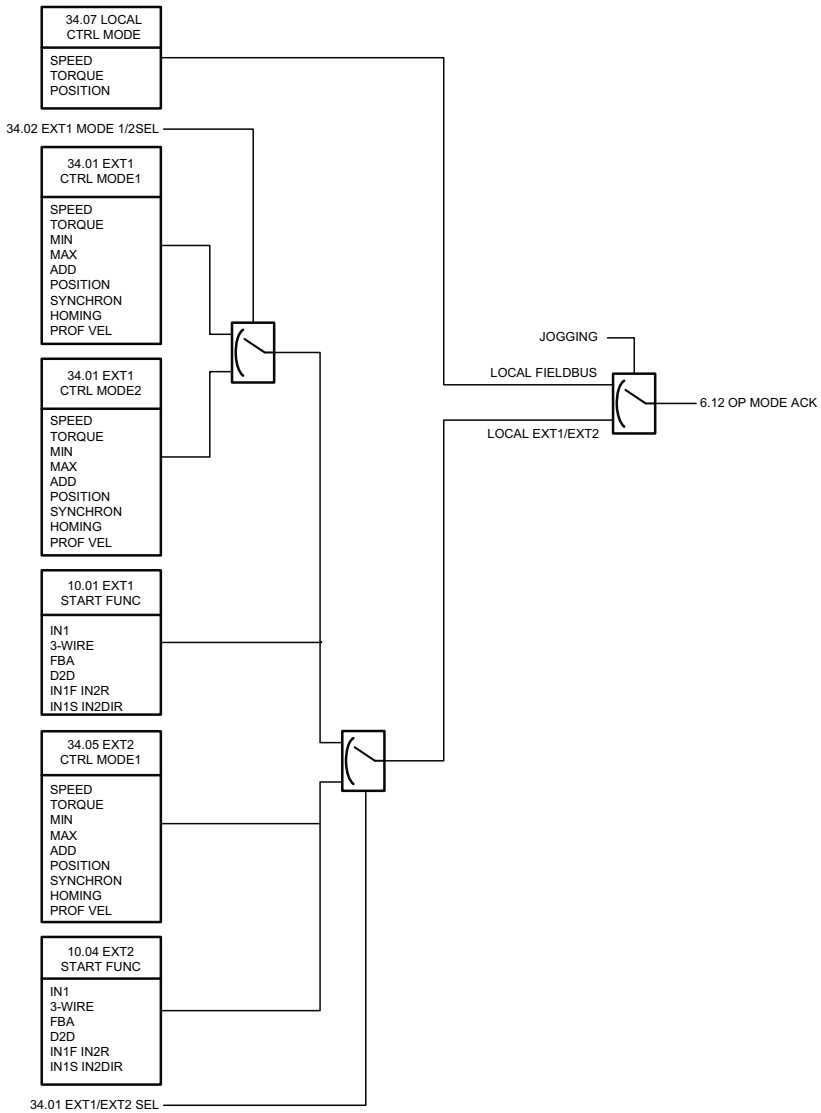
33.09	SUPERV3 FUNC	FW block: SUPERVISION (see above)
	Selects the mode of supervision 3.	
	(0) Disabled	Supervision 3 not in use.
	(1) Low	When the signal selected by parameter 33.10 SUPERV3 ACT falls below the value of parameter 33.12 SUPERV3 LIM LO, bit 2 of 6.14 SUPERV STATUS is activated. To clear the bit, the signal must exceed the value of parameter 33.11 SUPERV3 LIM HI.
	(2) High	When the signal selected by parameter 33.10 SUPERV3 ACT exceeds the value of parameter 33.11 SUPERV3 LIM HI, bit 2 of 6.14 SUPERV STATUS is activated. To clear the bit, the signal must fall below the value of parameter 33.12 SUPERV3 LIM LO.
	(3) Abs Low	When the absolute value of the signal selected by parameter 33.10 SUPERV3 ACT falls below the value of parameter 33.12 SUPERV3 LIM LO, bit 2 of 6.14 SUPERV STATUS is activated. To clear the bit, the absolute value of the signal must exceed the value of parameter 33.11 SUPERV3 LIM HI.
	(4) Abs High	When the absolute value of the signal selected by parameter 33.10 SUPERV3 ACT exceeds the value of parameter 33.11 SUPERV3 LIM HI, bit 2 of 6.14 SUPERV STATUS is activated. To clear the bit, the absolute value of the signal must fall below the value of parameter 33.12 SUPERV3 LIM LO.
33.10	SUPERV3 ACT	FW block: SUPERVISION (see above)
	Selects the signal to be monitored by supervision 3. See parameter 33.09 SUPERV3 FUNC.	
	Value pointer: Group and index	
33.11	SUPERV3 LIM HI	FW block: SUPERVISION (see above)
	Sets the upper limit for supervision 3. See parameter 33.09 SUPERV3 FUNC.	
	-32768…32768	Upper limit for supervision 3.
33.12	SUPERV3 LIM LO	FW block: SUPERVISION (see above)
	Sets the lower limit for supervision 3. See parameter 33.09 SUPERV3 FUNC.	
	-32768…32768	Lower limit for supervision 3.

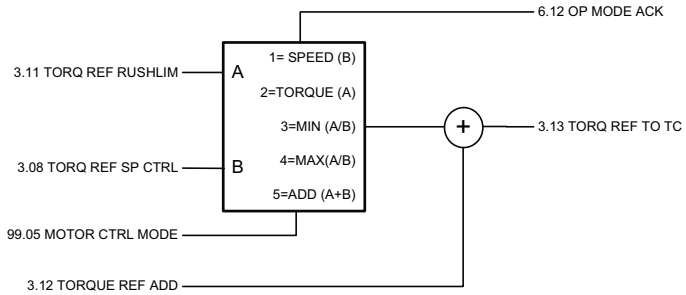
Group 34 REFERENCE CTRL

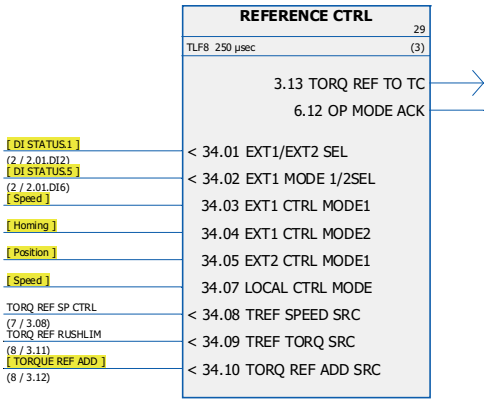
Reference source and type selection.

Using the parameters in this group, it is possible to select whether external control location EXT1 or EXT2 is used (either one is active at a time). These parameters also select the control mode (SPEED/TORQUE/MIN/MAX/ADD) and the used torque reference in local and external control. It is possible to select two different control modes for the EXT1 location by using parameters 34.03 EXT1 CTRL MODE1 and 34.04 EXT1 CTRL MODE2; the same start/stop commands are used in both modes.

For more information on control locations and control modes, see chapter *Drive control and features*. For start/stop control in different control locations, see parameter group 10 START/STOP (page 78).





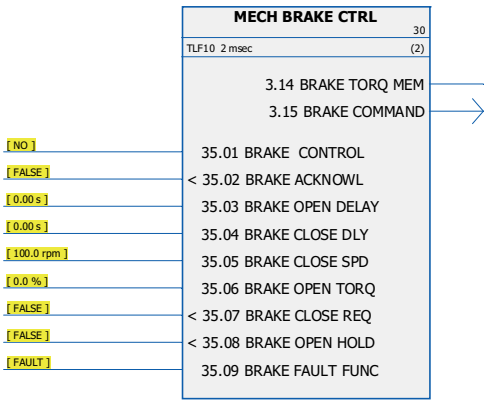
Firmware block: **REFERENCE CTRL** (34) This block • defines the selection method between external control locations EXT1 and EXT2 • configures control mode (SPEED/ TORQUE/MIN/MAX/ADD) selection • selects the torque reference used in local and external control • shows the torque reference (for torque control) and operating mode.		
Block outputs located in other parameter groups		3.13 TORQ REF TO TC (page 68) 6.12 OP MODE ACK (page 73)
34.01	EXT1/EXT2 SEL	FW block: REFERENCE CTRL (see above)
	Selects the source for external control location EXT1/EXT2 selection. 0 = EXT1. 1 = EXT2.	
	Bit pointer: Group, index and bit	
34.02	EXT1 MODE 1/2SEL	FW block: REFERENCE CTRL (see above)
	Selects the source for EXT1 control mode 1/2 selection. 1 = mode 2. 0 = mode 1. Control mode 1/2 is selected by parameter 34.03 EXT1 CTRL MODE1 / 34.04 EXT1 CTRL MODE2.	
	Bit pointer: Group, index and bit	
34.03	EXT1 CTRL MODE1	FW block: REFERENCE CTRL (see above)
	Selects control mode 1 for external control location EXT1.	
	(1) Speed	Speed control. Torque reference is 3.08 TORQ REF SP CTRL, which is the output of the SPEED CONTROL firmware block. Torque reference source can be changed by parameter 34.08 TREF SPEED SRC.

	(2) Torque	Torque control. Torque reference is 3.11 TORQ REF RUSHLIM, which is the output of the TORQ REF MOD firmware block. Torque reference source can be changed by parameter 34.09 TREF TORQ SRC.
	(3) Min	Combination of selections (1) Speed and (2) Torque: Torque selector compares the torque reference and the speed controller output and the smaller of them is used.
	(4) Max	Combination of selections (1) Speed and (2) Torque: Torque selector compares the torque reference and the speed controller output and the greater of them is used.
	(5) Add	Combination of selections (1) Speed and (2) Torque: Torque selector adds the speed controller output to the torque reference.
34.04	EXT1 CTRL MODE2	FW block: REFERENCE CTRL (see above)
	Selects control mode 2 for external control location EXT1. For selections, see parameter 34.03 EXT1 CTRL MODE1.	
34.05	EXT2 CTRL MODE1	FW block: REFERENCE CTRL (see above)
	Selects control mode for external control location EXT2. For selections, see parameter 34.03 EXT1 CTRL MODE1.	
34.07	LOCAL CTRL MODE	FW block: REFERENCE CTRL (see above)
	Selects the control mode for local control. **Note:** This parameter cannot be changed while the drive is running.	
	(1) Speed	Speed control. Torque reference is 3.08 TORQ REF SP CTRL, which is the output of the SPEED CONTROL firmware block. Torque reference source can be changed by parameter 34.08 TREF SPEED SRC.
	(2) Torque	Torque control. Torque reference is 3.11 TORQ REF RUSHLIM, which is an output of the TORQ REF MOD firmware block. Torque reference source can be changed by parameter 34.09 TREF TORQ SRC.
34.08	TREF SPEED SRC	FW block: REFERENCE CTRL (see above)
	Selects the source for the torque reference (from the speed controller). Default value is P.3.8, i.e. 3.08 TORQ REF SP CTRL, which is the output of the SPEED CONTROL firmware block. **Note:** This parameter has been locked, i.e. no user setting is possible.	
	Value pointer: Group and index	
34.09	TREF TORQ SRC	FW block: REFERENCE CTRL (see above)
	Selects the source for the torque reference (from the torque reference chain). Default value is P.3.11, i.e. 3.11 TORQ REF RUSHLIM, which is an output of the TORQ REF MOD firmware block. **Note:** This parameter has been locked, i.e. no user setting is possible.	
	Value pointer: Group and index	

34.10	TORQ REF ADD SRC	FW block: REFERENCE CTRL (see above)
	Selects the source for the torque reference added to the torque value after the torque selection. Default value is P.3.12, i.e. 3.12 TORQUE REF ADD, which is an output of the TORQ REF SEL firmware block. **Note:** This parameter has been locked, i.e. no user setting is possible.	
	Value pointer: Group and index	

Group 35 MECH BRAKE CTRL

Settings for the control of a mechanical brake. See also section *Mechanical brake control* on page 49.

Firmware block: **MECH BRAKE CTRL** (35)		
Block outputs located in other parameter groups		3.14 BRAKE TORQ MEM (page 68) 3.15 BRAKE COMMAND (page 69)
35.01	BRAKE CONTROL	FW block: MECH BRAKE CTRL (see above)
	Activates the brake control function with or without supervision. **Note:** This parameter cannot be changed while the drive is running.	
	(0) NO	Inactive.
	(1) WITH ACK	Brake control with supervision (supervision is activated by parameter 35.02 BRAKE ACKNOWL).
	(2) NO ACK	Brake control without supervision.
35.02	BRAKE ACKNOWL	FW block: MECH BRAKE CTRL (see above)
	Selects the source for the external brake on/off supervision activation (when par. 35.01 BRAKE CONTROL = (1) WITH ACK). The use of the external on/off supervision signal is optional. 1 = The brake is open. 0 = The brake is closed. Brake supervision is usually controlled with a digital input. It can also be controlled with an external control system, e.g. fieldbus. When brake control error is detected the drive reacts as defined by parameter 35.09 BRAKE FAULT FUNC. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	

35.03	BRAKE OPEN DELAY	FW block: MECH BRAKE CTRL (see above)
	Defines the brake open delay (= the delay between the internal open brake command and the release of the motor speed control). The delay counter starts when the drive has magnetised the motor and risen the motor torque to the level required at the brake release (parameter 35.06 BRAKE OPEN TORQ). Simultaneously with the counter start, the brake function energises the relay output controlling the brake and the brake starts opening. Set the delay the same as the mechanical opening delay of the brake specified by the brake manufacturer.	
	0…5 s	Brake open delay.
35.04	BRAKE CLOSE DLY	FW block: MECH BRAKE CTRL (see above)
	Defines the brake close delay. The delay counter starts when the motor actual speed has fallen below the set level (parameter 35.05 BRAKE CLOSE SPD) after the drive has received the stop command. Simultaneously with the counter start, the brake control function de-energises the relay output controlling the brake and the brake starts closing. During the delay, the brake function keeps the motor live preventing the motor speed from falling below zero. Set the delay time to the same value as the mechanical make-up time of the brake (= operating delay when closing) specified by the brake manufacturer.	
	0…60 s	Brake close delay.
35.05	BRAKE CLOSE SPD	FW block: MECH BRAKE CTRL (see above)
	Defines the brake close speed (an absolute value). See parameter 35.04 BRAKE CLOSE DLY.	
	0…1000 rpm	Brake close speed.
35.06	BRAKE OPEN TORQ	FW block: MECH BRAKE CTRL (see above)
	Defines the motor starting torque at brake release (in percent of the motor nominal torque).	
	0…1000%	Motor starting torque at brake release.
35.07	BRAKE CLOSE REQ	FW block: MECH BRAKE CTRL (see above)
	Selects the source for the brake close (open) request. 1 = Brake close request. 0 = Brake open request. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
35.08	BRAKE OPEN HOLD	FW block: MECH BRAKE CTRL (see above)
	Selects the source for the activation of the brake open command hold. 1 = Hold active. 0 = Normal operation. **Note:** This parameter cannot be changed while the drive is running.	
	Bit pointer: Group, index and bit	
35.09	BRAKE FAULT FUNC	FW block: MECH BRAKE CTRL (see above)
	Defines how the drive reacts in case of mechanical brake control error. If brake control supervision has not been activated by parameter 35.01 BRAKE CONTROL, this parameter is disabled.	

	(0) FAULT	The drive trips on fault BRAKE NOT CLOSED / BRAKE NOT OPEN if the status of the optional external brake acknowledgement signal does not meet the status presumed by the brake control function. The drive trips on fault BRAKE START TORQUE if the required motor starting torque at brake release is not achieved.
	(1) ALARM	The drive generates alarm BRAKE NOT CLOSED / BRAKE NOT OPEN if the status of the optional external brake acknowledgement signal does not meet the status presumed by the brake control function. The drive generates alarm BRAKE START TORQUE if the required motor starting torque at brake release is not achieved.
	(2) OPEN FLT	The drive generates alarm BRAKE NOT CLOSED (when closing the brake) and trips on fault BRAKE NOT OPEN (when opening the brake) if the status of the optional external brake acknowledgement signal does not match the status presumed by the brake control function. The drive trips on BRAKE START TORQUE if the required motor start torque at brake release is not achieved.

Group 40 MOTOR CONTROL

Motor control settings, such as

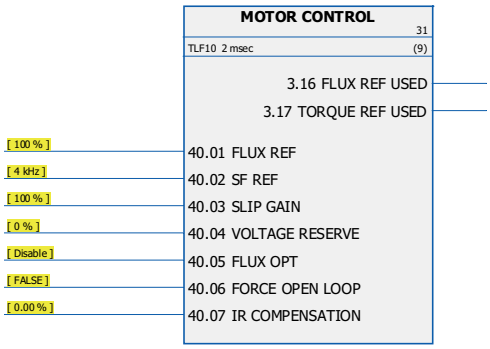
- flux reference
- drive switching frequency
- motor slip compensation
- voltage reserve
- flux optimisation
- IR compensation for scalar control mode.

Flux optimisation

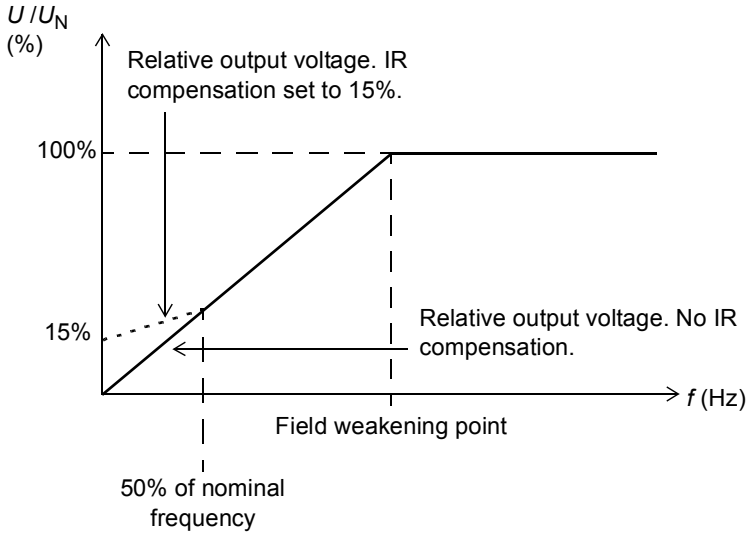
Flux optimisation reduces the total energy consumption and motor noise level when the drive operates below the nominal load. The total efficiency (motor and drive) can be improved by 1% to 10%, depending on the load torque and speed.

Note: Flux optimisation limits the dynamic control performance of the drive because with a small flux reference the drive torque cannot be increased fast.

<table>
<tr><td colspan="2">Firmware block:
MOTOR CONTROL
(40)

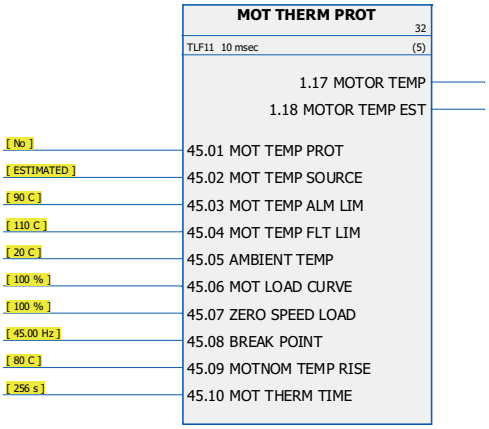
This block defines motor control settings such as
• flux reference
• drive switching frequency
• motor slip compensation
• voltage reserve
• flux optimisation
• IR compensation for scalar control mode.
The block also shows the flux and torque reference used.</td><td> 
</td></tr>
<tr><td colspan="2">Block outputs located in other parameter groups</td><td>3.16 FLUX REF USED (page 69)
3.17 TORQUE REF USED (page 69)</td></tr>
<tr><td>40.01</td><td>FLUX REF</td><td>FW block: MOTOR CONTROL (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the flux reference.</td></tr>
<tr><td></td><td>0…200%</td><td>Flux reference.</td></tr>
<tr><td>40.02</td><td>SF REF</td><td>FW block: MOTOR CONTROL (see above)</td></tr>
<tr><td></td><td colspan="2">Defines the switching frequency of the drive.
When switching frequency exceeds 4 kHz, the allowed drive output current is limited. See switching frequency derating in the appropriate Hardware Manual.</td></tr>
<tr><td></td><td>1/2/3/4/5/8/16 kHz</td><td>Switching frequency.</td></tr>
</table>

40.03	SLIP GAIN	FW block: MOTOR CONTROL (see above)
	Defines the slip gain which is used to improve the estimated motor slip. 100% means full slip gain; 0% means no slip gain. The default value is 100%. Other values can be used if a static speed error is detected despite of the full slip gain. Example (with nominal load and nominal slip of 40 rpm): A 1000 rpm constant speed reference is given to the drive. Despite of the full slip gain (= 100%), a manual tachometer measurement from the motor axis gives a speed value of 998 rpm. The static speed error is 1000 rpm - 998 rpm = 2 rpm. To compensate the error, the slip gain should be increased. At the 105% gain value, no static speed error exists (2 rpm / 40 rpm = 5%).	
	0…200%	Slip gain.
40.04	VOLTAGE RESERVE	FW block: MOTOR CONTROL (see above)
	Defines the minimum allowed voltage reserve. When the voltage reserve has decreased to the set value, the drive enters the field weakening area. If the intermediate circuit DC voltage U_{dc} = 550 V and the voltage reserve is 5%, the RMS value of the maximum output voltage in steady-state operation is 0.95 × 550 V / sqrt(2) = 369 V The dynamic performance of the motor control in the field weakening area can be improved by increasing the voltage reserve value, but the drive enters the field weakening area earlier.	
	-4…50 %	Minimum allowed voltage reserve.
40.05	FLUX OPT	FW block: MOTOR CONTROL (see above)
	Enables the flux optimisation function. Flux optimisation improves motor efficiency and reduces noise. Flux optimisation is used in drives that usually operate below nominal load. **Note:** With a permanent magnet motor, flux optimisation is always enabled regardless of this parameter.	
	(0) Disable	Flux optimisation disabled.
	(1) Enable	Flux optimisation enabled.
40.06	FORCE OPEN LOOP	FW block: MOTOR CONTROL (see above)
	Defines the speed/position information used by the motor model.	
	(0) FALSE	Motor model uses the speed feedback selected by parameter 22.01 SPEED FB SEL.
	(1) TRUE	Motor model uses the internal speed estimate (even when parameter 22.01 SPEED FB SEL setting is (1) Enc1 speed / (2) Enc2 speed).

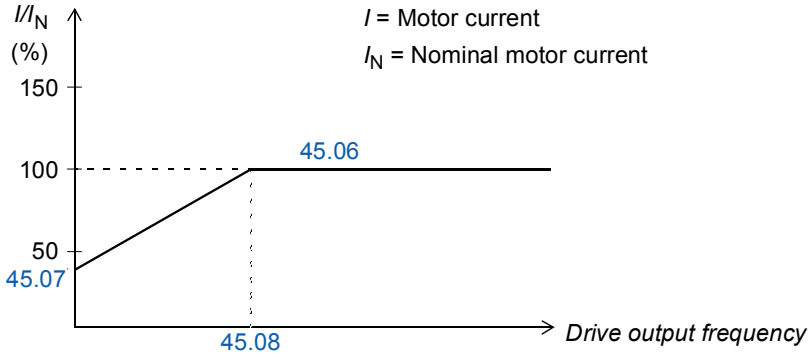
40.07	IR COMPENSATION	FW block: MOTOR CONTROL (see above)
	Defines the relative output voltage boost at zero speed (IR compensation). The function is useful in applications with high break-away torque when no DTC motor can be applied. This parameter is only effective when parameter 99.05 MOTOR CTRL MODE is set to (1) Scalar. 	
	0…50%	IR compensation.

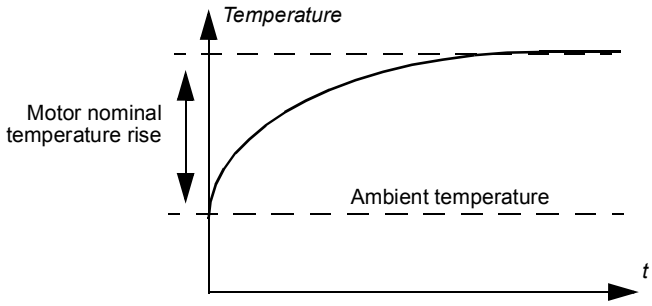
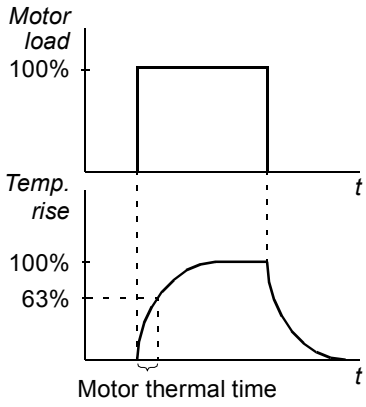
Group 45 MOT THERM PROT

Settings for thermal protection of the motor. See also section *Thermal motor protection* on page 39.

Firmware block: **MOT THERM PROT** (45) Configures motor overtemperature protection and temperature measurement. Also shows the estimated and measured motor temperatures.		
Block outputs located in other parameter groups		1.17 MOTOR TEMP (page 60) 1.18 MOTOR TEMP EST (page 60)
45.01	MOT TEMP PROT	FW block: MOT THERM PROT (see above)
	Selects how the drive reacts when motor overtemperature is detected.	
	(0) No	Inactive.
	(1) Alarm	The drive generates alarm MOTOR TEMPERATURE when the temperature exceeds the alarm level defined by parameter 45.03 MOT TEMP ALM LIM.
	(2) Fault	The drive generates alarm MOTOR TEMPERATURE or trips on fault MOTOR OVERTEMP when the temperature exceeds the alarm/fault level defined by parameter 45.03 MOT TEMP ALM LIM / 45.04 MOT TEMP FLT LIM.
45.02	MOT TEMP SOURCE	FW block: MOT THERM PROT (see above)
	Selects the motor temperature protection. When overtemperature is detected the drive reacts as defined by parameter 45.01 MOT TEMP PROT.	
	(0) ESTIMATED	The temperature is supervised based on the motor thermal protection model, which uses the motor thermal time constant (parameter 45.10 MOT THERM TIME) and the motor load curve (parameters 45.06…45.08). User tuning is typically needed only if the ambient temperature differs from the normal operating temperature specified for the motor. The motor temperature increases if it operates in the region above the motor load curve. The motor temperature decreases if it operates in the region below the motor load curve (if the motor is overheated). **WARNING!** The model does not protect the motor if it does not cool properly due to dust and dirt.

	(1) KTY JCU	The temperature is supervised using a KTY84 sensor connected to drive thermistor input TH.
	(2) KTY 1st FEN	The temperature is supervised using a KTY84 sensor connected to encoder interface module FEN-xx installed in drive Slot 1/2. If two encoder interface modules are used, encoder module connected to Slot 1 is used for the temperature supervision. **Note:** This selection does not apply for FEN-01. *
	(3) KTY 2nd FEN	The temperature is supervised using a KTY84 sensor connected to encoder interface module FEN-xx installed in drive Slot 1/2. If two encoder interface modules are used, encoder module connected to Slot 2 is used for the temperature supervision. **Note:** This selection does not apply for FEN-01. *
	(4) PTC JCU	The temperature is supervised using 1…3 PTC sensors connected to drive thermistor input TH.
	(5) PTC 1st FEN	The temperature is supervised using a PTC sensor connected to encoder interface module FEN-xx installed in drive Slot 1/2. If two encoder interface modules are used, encoder module connected to Slot 1 is used for the temperature supervision. *
	(6) PTC 2nd FEN	The temperature is supervised using a PTC sensor connected to encoder interface module FEN-xx installed in drive Slot 1/2. If two encoder interface modules are used, encoder module connected to Slot 2 is used for the temperature supervision. *
	***Note:** If one FEN-xx module is used, parameter setting must be either (2) KTY 1st FEN or (5) PTC 1st FEN. The FEN-xx module can be in either Slot 1 or Slot 2.	
45.03	MOT TEMP ALM LIM	FW block: MOT THERM PROT (see above)
	Defines the alarm limit for the motor overtemperature protection (when par. 45.01 MOT TEMP PROT = (1) Alarm or (2) Fault).	
	0…200 °C	Motor overtemperature alarm limit.
45.04	MOT TEMP FLT LIM	FW block: MOT THERM PROT (see above)
	Defines the fault limit for the motor overtemperature protection (when par. 45.01 MOT TEMP PROT = (2) Fault).	
	0…200 °C	Motor overtemperature fault limit.
45.05	AMBIENT TEMP	FW block: MOT THERM PROT (see above)
	Defines the ambient temperature for the thermal protection mode.	
	-60…100 °C	Ambient temperature.

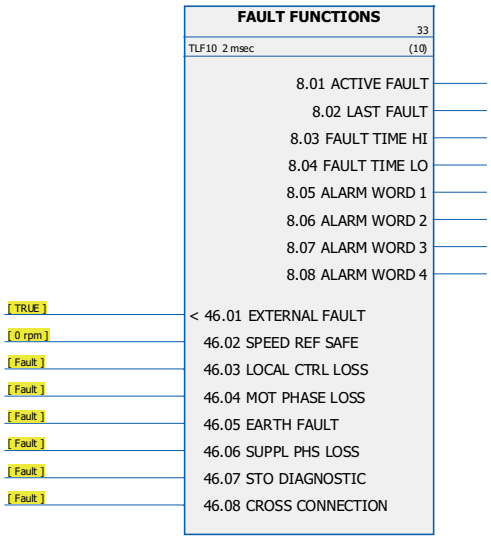
45.06	MOT LOAD CURVE	FW block: MOT THERM PROT (see above)
	Defines the load curve together with parameters 45.07 ZERO SPEED LOAD and 45.08 BREAK POINT. The value is given in percent of nominal motor current. When the parameter is set to 100%, the maximum load is equal to the value of the parameter 99.06 MOT NOM CURRENT (higher loads heat up the motor). The load curve level should be adjusted if the ambient temperature differs from the nominal value.  The load curve is used by the motor thermal protection model when parameter 45.02 MOT TEMP SOURCE is set to (0) ESTIMATED.	
	50…150%	Motor current above breakpoint.
45.07	ZERO SPEED LOAD	FW block: MOT THERM PROT (see above)
	Defines the load curve together with parameters 45.06 MOT LOAD CURVE and 45.08 BREAK POINT. Defines the maximum motor load at zero speed of the load curve. A higher value can be used if the motor has an external motor fan to boost the cooling. See the motor manufacturer's recommendations. The value is given in percent of nominal motor current. The load curve is used by the motor thermal protection model when parameter 45.02 MOT TEMP SOURCE is set to (0) ESTIMATED.	
	50…150%	Motor current at zero speed.
45.08	BREAK POINT	FW block: MOT THERM PROT (see above)
	Defines the load curve together with parameters 45.06 MOT LOAD CURVE and 45.07 ZERO SPEED LOAD. Defines the break point frequency of the load curve i.e. the point at which the motor load curve begins to decrease from the value of parameter 45.06 MOT LOAD CURVE to the value of parameter 45.07 ZERO SPEED LOAD. The load curve is used by the motor thermal protection model when parameter 45.02 MOT TEMP SOURCE is set to (0) ESTIMATED.	
	0.01…500 Hz	Load curve breakpoint.

45.09	MOTNOM TEMP RISE	FW block: MOT THERM PROT (see above)
	Defines the temperature rise of the motor when the motor is loaded with nominal current. See the motor manufacturer's recommendations. The temperature rise value is used by the motor thermal protection model when parameter 45.02 MOT TEMP SOURCE is set to (0) ESTIMATED.  	
	0…300 °C	Motor temperature rise.
45.10	MOT THERM TIME	FW block: MOT THERM PROT (see above)
	Defines the thermal time constant for the motor thermal protection model (i.e. time inside which the temperature has reached 63% of the nominal temperature). See the motor manufacturer's recommendations. The motor thermal protection model is used when parameter 45.02 MOT TEMP SOURCE is set to (0) ESTIMATED.  	
	100…10000 s	Motor thermal time.

Group 46 FAULT FUNCTIONS

Definition of drive behaviour upon a fault situation.

An alarm or a fault message indicates abnormal drive status. For the possible causes and remedies, see chapter *Fault tracing*.

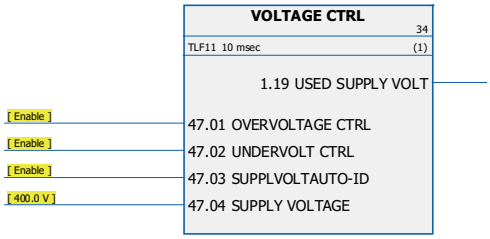
Firmware block: **FAULT FUNCTIONS** (46) This block • configures supervision of external faults by defining the source (for example, a digital input) for external fault indication signal • selects the reaction of the drive (alarm; fault; continuation at safe speed in some cases) upon situations like local control communication break, motor/supply phase loss, earth fault, or Safe Torque Off function activation • shows the codes of the latest faults, the time at which the active fault occurred, and the alarm words.		
Block outputs located in other parameter groups		8.01 ACTIVE FAULT (page 74) 8.02 LAST FAULT (page 74) 8.03 FAULT TIME HI (page 74) 8.04 FAULT TIME LO (page 74) 8.05 ALARM WORD 1 (page 74) 8.06 ALARM WORD 2 (page 75) 8.07 ALARM WORD 3 (page 75) 8.08 ALARM WORD 4 (page 76)
46.01	EXTERNAL FAULT	FW block: FAULT FUNCTIONS (see above)
	Selects an interface for an external fault signal. 0 = External fault trip. 1 = No external fault.	
	Bit pointer: Group, index and bit	
46.02	SPEED REF SAFE	FW block: FAULT FUNCTIONS (see above)
	Defines the fault speed. Used as a speed reference when an alarm occurs when parameter 13.12 AI SUPERVISION / 46.03 LOCAL CTRL LOSS / 50.02 COMM LOSS FUNC setting is (2) Spd ref Safe.	
	-30000…30000 rpm	Fault speed.
46.03	LOCAL CTRL LOSS	FW block: FAULT FUNCTIONS (see above)
	Selects how the drive reacts to a control panel or PC tool communication break.	
	(0) No	No action.
	(1) Fault	Drive trips on LOCAL CTRL LOSS fault.

	(2) Spd ref Safe	The drive generates alarm LOCAL CTRL LOSS and sets the speed to the speed defined by parameter 46.02 SPEED REF SAFE. **WARNING!** Make sure that it is safe to continue operation in case of a communication break.
	(3) Last speed	The drive generates alarm LOCAL CTRL LOSS and freezes the speed to the level the drive was operating at. The speed is determined by the average speed over the previous 10 seconds. **WARNING!** Make sure that it is safe to continue operation in case of a communication break.
46.04	MOT PHASE LOSS	FW block: FAULT FUNCTIONS (see above)
	Selects how the drive reacts when a motor phase loss is detected.	
	(0) No	No action.
	(1) Fault	Drive trips on MOTOR PHASE fault.
46.05	EARTH FAULT	FW block: FAULT FUNCTIONS (see above)
	Selects how the drive reacts when an earth fault or current unbalance is detected in the motor or the motor cable.	
	(0) No	No action.
	(1) Warning	Drive generates alarm EARTH FAULT.
	(2) Fault	Drive trips on EARTH FAULT.
46.06	SUPPL PHS LOSS	FW block: FAULT FUNCTIONS (see above)
	Selects how the drive reacts when a supply phase loss is detected.	
	(0) No	No reaction.
	(1) Fault	Drive trips on SUPPLY PHASE fault.
46.07	STO DIAGNOSTIC	FW block: FAULT FUNCTIONS (see above)
	Selects how the drive reacts when it detects the absence of one or both Safe Torque Off (STO) signals. **Note:** This parameter is for supervision only. The Safe Torque Off function can activate even when this parameter is set to (3) No. For general information on the Safe Torque Off function, see the *Hardware Manual* of the drive.	
	(1) Fault	The drive trips on SAFE TORQUE OFF when one or both of the STO signals are lost.
	(2) Alarm	Drive running: The drive trips on SAFE TORQUE OFF when one or both of the STO signals is lost. Drive stopped: The drive generates a SAFE TORQUE OFF alarm if both STO signals are absent. If only one of the signals is lost, the drive trips on STO1 LOST or STO2 LOST.

	(3) No	Drive running: The drive trips on SAFE TORQUE OFF when one or both of the STO signals is lost. Drive stopped: No action if both STO signals are absent. If only one of the signals is lost, the drive trips on STO1 LOST or STO2 LOST.
	(4) Only Alarm	The drive generates a SAFE TORQUE OFF alarm if both STO signals are absent. If only one of the signals is lost, the drive trips on STO1 LOST or STO2 LOST.
46.08	CROSS CONNECTION	FW block: FAULT FUNCTIONS (see above)
	Selects how the drive reacts to incorrect input power and motor cable connection (i.e. input power cable is connected to drive motor connection).	
	(0) No	No reaction.
	(1) Fault	Drive trips on CABLE CROSS CON fault.

Group 47 VOLTAGE CTRL

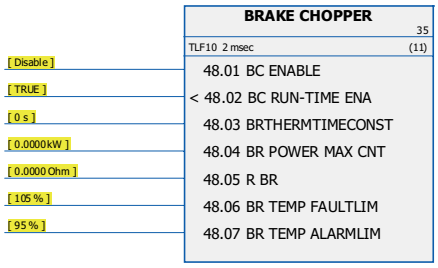
Settings for overvoltage and undervoltage control, and supply voltage.

Firmware block: **VOLTAGE CTRL** (47) This block • enables/disables overvoltage and undervoltage control • enables/disables automatic identification of supply voltage • provides a parameter for manual definition of supply voltage • shows the supply voltage value used by the control program.		
Block outputs located in other parameter groups		1.19 USED SUPPLY VOLT (page 60)
47.01	OVERVOLTAGE CTRL	FW block: VOLTAGE CTRL (see above)
	Enables the overvoltage control of the intermediate DC link. Fast braking of a high inertia load causes the voltage to rise to the overvoltage control limit. To prevent the DC voltage from exceeding the limit, the overvoltage controller automatically decreases the braking torque. **Note:** If a brake chopper and resistor or a regenerative supply section are included in the drive, the controller must be disabled.	
	(0) Disable	Overvoltage control disabled.
	(1) Enable	Overvoltage control enabled.
47.02	UNDERVOLT CTRL	FW block: VOLTAGE CTRL (see above)
	Enables the undervoltage control of the intermediate DC link. If the DC voltage drops due to input power cut off, the undervoltage controller will automatically decrease the motor torque in order to keep the voltage above the lower limit. By decreasing the motor torque, the inertia of the load will cause regeneration back to the drive, keeping the DC link charged and preventing an undervoltage trip until the motor coasts to stop. This will act as a power-loss ride-through functionality in systems with high inertia, such as a centrifuge or a fan.	
	(0) Disable	Undervoltage control disabled.
	(1) Enable	Undervoltage control enabled.
47.03	SUPPLVOLTAUTO-ID	FW block: VOLTAGE CTRL (see above)
	Enables the auto-identification of the supply voltage. See also section *Voltage control and trip limits* on page 42.	
	(0) Disable	Auto-identification of supply voltage disabled.
	(1) Enable	Auto-identification of supply voltage enabled.

47.04	SUPPLY VOLTAGE	FW block: VOLTAGE CTRL (see above)
	Defines the nominal supply voltage. Used if auto-identification of the supply voltage is not enabled by parameter 47.03 SUPPLVOLTAUTO-ID.	
	0…1000 V	Nominal supply voltage.
47.05	LOW VOLT MOD ENA	FW block: None
	Enables/disables (or selects a signal source that enables/disables) Low voltage mode. 0 = Low voltage mode disabled, 1 = Low voltage mode enabled. See section *Low voltage mode* on page 43.	
	Bit pointer: Group, index and bit	
47.06	LOW VOLT DC MIN	FW block: None
	Minimum DC voltage for Low voltage mode. See section *Low voltage mode* on page 43.	
	250…450 V	Minimum DC voltage for Low voltage mode.
47.07	LOW VOLT DC MAX	FW block: None
	Maximum DC voltage for Low voltage mode. See section *Low voltage mode* on page 43. **Note:** The value of this parameter must be higher than (47.06 LOW VOLT DC MIN + 50 V).	
	350…810 V	Maximum DC voltage for Low voltage mode.
47.08	EXT PU SUPPLY	FW block: None
	Enables/disables (or selects a signal source that enables/disables) external power unit supply, used with low DC supply voltages such as a battery. 0 = External power unit supply disabled, 1 = External power unit supply enabled. See section *Low voltage mode* on page 43.	
	Bit pointer: Group, index and bit	

Group 48 BRAKE CHOPPER

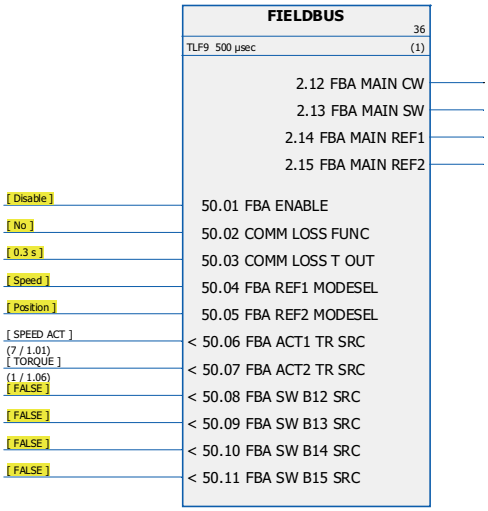
Configuration of internal brake chopper.

Firmware block: **BRAKE CHOPPER** (48) This block configures the brake chopper control and supervision.		
48.01	BC ENABLE	FW block: BRAKE CHOPPER (see above)
	Enables the brake chopper control. **Note:** Before enabling brake chopper control, ensure the brake resistor is installed and the overvoltage control is switched off (parameter 47.01 OVERVOLTAGE CTRL). The drive has a built-in brake chopper.	
	(0) Disable	Brake chopper control disabled.
	(1) EnableTherm	Enable brake chopper control with resistor overload protection.
	(2) Enable	Enable brake chopper control without resistor overload protection. This setting can be used, for example, if the resistor is equipped with a thermal circuit breaker that is wired to stop the drive if the resistor overheats.
48.02	BC RUN-TIME ENA	FW block: BRAKE CHOPPER (see above)
	Selects the source for quick run-time brake chopper control. 0 = Brake chopper IGBT pulses are cut off. 1 = Normal brake chopper IGBT modulation. The overvoltage control is automatically switched off. This parameter can be used to program the chopper control to function only when the drive is operating in generating mode.	
	Bit pointer: Group, index and bit	
48.03	BRTHERMTIMECONST	FW block: BRAKE CHOPPER (see above)
	Defines the thermal time constant of the brake resistor for overload protection.	
	0…10000 s	Brake resistor thermal time constant.
48.04	BR POWER MAX CNT	FW block: BRAKE CHOPPER (see above)
	Defines the maximum continuous braking power which will raise the resistor temperature to the maximum allowed value. The value is used in the overload protection.	
	0…10000 kW	Maximum continuous braking power.
48.05	R BR	FW block: BRAKE CHOPPER (see above)
	Defines the resistance value of the brake resistor. The value is used for brake chopper protection.	

	0.1…1000 ohm	Resistance.
48.06	BR TEMP FAULTLIM	FW block: BRAKE CHOPPER (see above)
	Selects the fault limit for the brake resistor temperature supervision. The value is given in percent of the temperature the resistor reaches when loaded with the power defined by parameter 48.04 BR POWER MAX CNT. When the limit is exceeded the drive trips on fault BR OVERHEAT.	
	0…150%	Resistor temperature fault limit.
48.07	BR TEMP ALARMLIM	FW block: BRAKE CHOPPER (see above)
	Selects the alarm limit for the brake resistor temperature supervision. The value is given in percent of the temperature the resistor reaches when loaded with the power defined by parameter 48.04 BR POWER MAX CNT. When limit is exceeded the drive generates alarm BR OVERHEAT.	
	0…150%	Resistor temperature alarm limit.

Group 50 FIELDBUS

Basic settings for fieldbus communication. See also *Appendix A – Fieldbus control* on page 325.

Firmware block: **FIELDBUS** (50) This block • initialises the fieldbus communication • selects communication supervision method • defines scaling of the fieldbus references and actual values • selects sources for programmable status word bits • shows the fieldbus control and status words, and references.		
Block outputs located in other parameter groups		2.12 FBA MAIN CW (page 63) 2.13 FBA MAIN SW (page 65) 2.14 FBA MAIN REF1 (page 66) 2.15 FBA MAIN REF2 (page 66)
50.01	FBA ENABLE	FW block: FIELDBUS (see above)
	Enables communication between the drive and fieldbus adapter.	
	(0) Disable	No communication.
	(1) Enable	Communication between drive and fieldbus adapter enabled.
50.02	COMM LOSS FUNC	FW block: FIELDBUS (see above)
	Selects how the drive reacts upon a fieldbus communication break. The time delay is defined by parameter 50.03 COMM LOSS T OUT.	
	(0) No	Communication break detection disabled.
	(1) Fault	Communication break detection active. Upon a communication break, the drive trips on fault FIELDBUS COMM and coasts to stop.
	(2) Spd ref Safe	Communication break detection active. Upon a communication break, the drive generates alarm FIELDBUS COMM and sets the speed to the value defined by parameter 46.02 SPEED REF SAFE. **WARNING**! Make sure that it is safe to continue operation in case of a communication break.

	(3) Last speed	Communication break detection active. Upon a communication break, the drive generates alarm FIELDBUS COMM and freezes the speed to the level the drive was operating at. The speed is determined by the average speed over the previous 10 seconds. **WARNING**! Make sure that it is safe to continue operation in case of a communication break.
50.03	COMM LOSS T OUT	FW block: FIELDBUS (see above)
	Defines the time delay before the action defined by parameter 50.02 COMM LOSS FUNC is taken. Time count starts when the link fails to update the message.	
	0.3…6553.5 s	Delay for fieldbus communication loss function.
50.04	FBA REF1 MODESEL	FW block: FIELDBUS (see above)
	Selects the fieldbus reference FBA REF1 scaling and the actual value, which is sent to the fieldbus (FBA ACT1).	
	(0) Raw data	No scaling (i.e. data is transmitted without scaling). Source for the actual value, which is sent to the fieldbus, is selected by parameter 50.06 FBA ACT1 TR SRC.
	(1) Torque	Fieldbus adapter module uses torque reference scaling. Torque reference scaling is defined by the used fieldbus profile (e.g. with ABB Drives Profile integer value 10000 corresponds to 100% torque value). Signal 1.06 TORQUE is sent to the fieldbus as an actual value. See the *User's Manual* of the appropriate fieldbus adapter module.
	(2) Speed	Fieldbus adapter module uses speed reference scaling. Speed reference scaling is defined by the used fieldbus profile (e.g. with ABB Drives Profile integer value 20000 corresponds to the value of parameter 25.02 SPEED SCALING). Signal 1.01 SPEED ACT is sent to the fieldbus as an actual value. See the *User's Manual* of the appropriate fieldbus adapter module.
	(5) Auto	One of the above selections is chosen automatically according to the currently active control mode. See parameter group 34 REFERENCE CTRL.
50.05	FBA REF2 MODESEL	FW block: FIELDBUS (see above)
	Selects the fieldbus reference FBA REF2 scaling. See parameter 50.04 FBA REF1 MODESEL.	
50.06	FBA ACT1 TR SRC	FW block: FIELDBUS (see above)
	Selects the source for fieldbus actual value 1 when parameter 50.04 FBA REF1 MODESEL / 50.05 FBA REF2 MODESEL is set to (0) Raw data.	
	Value pointer: Group and index	
50.07	FBA ACT2 TR SRC	FW block: FIELDBUS (see above)
	Selects the source for fieldbus actual value 2 when parameter 50.04 FBA REF1 MODESEL / 50.05 FBA REF2 MODESEL is set to (0) Raw data.	
	Value pointer: Group and index	

50.08	FBA SW B12 SRC	FW block: FIELDBUS (see above)
	Selects the source for freely programmable fieldbus status word bit 28 (2.13 FBA MAIN SW bit 28). Note that this functionality may not be supported by the fieldbus communication profile.	
	Bit pointer: Group, index and bit	
50.09	FBA SW B13 SRC	FW block: FIELDBUS (see above)
	Selects the source for freely programmable fieldbus status word bit 29 (2.13 FBA MAIN SW bit 29). Note that this functionality may not be supported by the fieldbus communication profile.	
	Bit pointer: Group, index and bit	
50.10	FBA SW B14 SRC	FW block: FIELDBUS (see above)
	Selects the source for freely programmable fieldbus status word bit 30 (2.13 FBA MAIN SW bit 30). Note that this functionality may not be supported by the fieldbus communication profile.	
	Bit pointer: Group, index and bit	
50.11	FBA SW B15 SRC	FW block: FIELDBUS (see above)
	Selects the source for freely programmable fieldbus status word bit 31 (2.13 FBA MAIN SW bit 31). Note that this functionality may not be supported by the fieldbus communication profile.	
	Bit pointer: Group, index and bit	

Group 51 FBA SETTINGS

Further fieldbus communication configuration. These parameters need to be set only if a fieldbus adapter module is installed. See also *Appendix A – Fieldbus control* on page 325.

Notes:

- This parameter group is presented in the *User's Manual* of the fieldbus adapter as parameter group 1 or A.
- The new settings will take effect when the drive is powered up the next time (before powering off the drive, wait at least 1 minute), or when parameter 51.27 FBA PAR REFRESH is activated.

51.01	FBA TYPE	FW block: None
	Displays the fieldbus protocol on the basis of the adapter module installed.	
	Not defined	Fieldbus adapter module not found (not properly connected, or disabled by parameter 50.01 FBA ENABLE).
	(Fieldbus protocol)	Fieldbus adapter for the stated protocol installed.
51.02	FBA PAR2	FW block: None
…	….	….
51.26	FBA PAR26	FW block: None
	Parameters 51.02…51.26 are adapter module-specific. For more information, see the *User's Manual* of the fieldbus adapter module. Note that not all of these parameters are necessarily used.	
51.27	FBA PAR REFRESH	FW block: None
	Validates any changed adapter module configuration parameter settings. After refreshing, the value reverts automatically to (0) DONE. **Note:** This parameter cannot be changed while the drive is running.	
	(0) DONE	Refreshing done.
	(1) REFRESH	Refreshing.
51.28	PAR TABLE VER	FW block: None
	Displays the parameter table revision of the fieldbus adapter module mapping file stored in the memory of the drive. In format xyz, where x = major revision number; y = minor revision number; z = correction number.	
51.29	DRIVE TYPE CODE	FW block: None
	Displays the drive type code of the fieldbus adapter module mapping file stored in the memory of the drive. Example: 520 = ACSM1 Speed and Torque Control Program.	

51.30	MAPPING FILE VER	FW block: None
	Displays the fieldbus adapter module mapping file revision stored in the memory of the drive. In hexadecimal format. Example: 0x107 = revision 1.07.	
51.31	D2FBA COMM STA	FW block: None
	Displays the status of the fieldbus adapter module communication.	
	(0) IDLE	Adapter not configured.
	(1) EXEC. INIT	Adapter initializing.
	(2) TIME OUT	A timeout has occurred in the communication between the adapter and the drive.
	(3) CONFIG ERROR	Adapter configuration error – the major or minor revision code of the common program revision in the fieldbus adapter module is not the revision required by the module (see par. 51.32 FBA COMM SW VER), or mapping file upload has failed more than three times.
	(4) OFF-LINE	Adapter is off-line.
	(5) ON-LINE	Adapter is on-line.
	(6) RESET	Adapter is performing a hardware reset.
51.32	FBA COMM SW VER	FW block: None
	Displays the common program revision of the adapter module. In format axyz, where a = major revision number, xy = minor revision numbers. z = correction letter. Example: 190A = revision 1.90A.	
51.33	FBA APPL SW VER	FW block: None
	Displays the application program revision of the adapter module. In format axyz, where: a = major revision number, xy = minor revision numbers, z = correction letter. Example: 190A = revision 1.90A.	

Group 52 FBA DATA IN

These parameters select the data to be sent by the drive to the fieldbus controller, and need to be set only if a fieldbus adapter module is installed. See also *Appendix A – Fieldbus control* on page 325.

Notes:

- This parameter group is presented in the *User's Manual* of the fieldbus adapter as parameter group 3 or C.
- The new settings will take effect when the drive is powered up the next time (before powering off the drive, wait at least 1 minute), or when parameter 51.27 FBA PAR REFRESH is activated.
- The maximum number of data words is protocol-dependent.

52.01	FBA DATA IN1	FW block: None
	Selects data to be transferred from the drive to the fieldbus controller.	
	0	Not in use.
	4	Status Word (16 bits).
	5	Actual value 1 (16 bits).
	6	Actual value 2 (16 bits).
	14	Status Word (32 bits).
	15	Actual value 1 (32 bits).
	16	Actual value 2 (32 bits).
	101…9999	Parameter index.
52.02	FBA DATA IN2	FW block: None
…	…	
52.12	FBA DATA IN12	FW block: None
	See 52.01 FBA DATA IN1.	

Group 53 FBA DATA OUT

These parameters select the data to be sent by the fieldbus controller to the drive, and need to be set only if a fieldbus adapter module is installed. See also *Appendix A – Fieldbus control* on page 325.

Notes:

- This parameter group is presented in the *User's Manual* of the fieldbus adapter as parameter group 2 or B.
- The new settings will take effect when the drive is powered up the next time (before powering off the drive, wait at least 1 minute), or when parameter 51.27 FBA PAR REFRESH is activated.
- The maximum number of data words is protocol-dependent.

53.01	FBA DATA OUT1	FW block: None
	Selects data to be transferred from the fieldbus controller to the drive.	
	0	Not in use.
	1	Control Word (16 bits).
	2	Reference REF1 (16 bits).
	3	Reference REF2 (16 bits).
	11	Control Word (32 bits).
	12	Reference REF1 (32 bits).
	13	Reference REF2 (32 bits).
	1001…9999	Parameter index.
53.02	FBA DATA OUT2	FW block: None
…		
53.12	FBA DATA OUT12	FW block: None
	See 53.01 FBA DATA OUT1.	

Group 55 COMMUNICATION TOOL

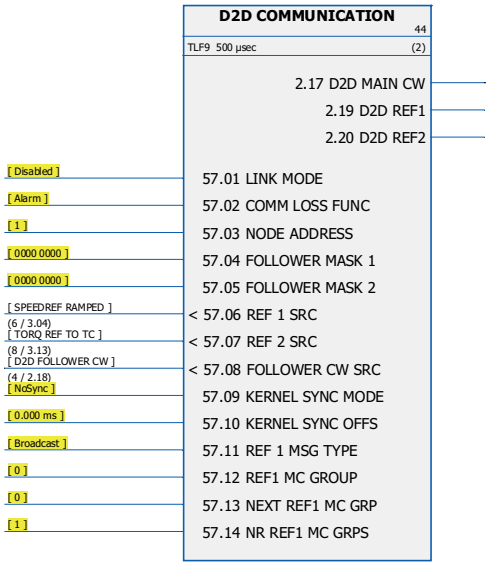
Settings for an RS-485 network implemented using optional JPC-01 Network communication adapters. The network enables the use of a single PC or control panel to control multiple drives.

For more information, see the *JPC-01 Network communication adapter User's manual* (3AUA0000072233).

55.01	MDB STATION ID	FW block: None
	Defines the ID of the drive on the RS-485 network. Each drive must have a dedicated ID number.	
	1…247	ID number. For drives, use a number between 1 and 31. (DriveStudio uses ID number 247.)
55.02	MDB BAUD RATE	FW block: None
	Sets the baud rate on the network. **Note:** This parameter must be set to (0) Auto if a control panel is used as the controlling device.	
	(0) Auto	Baud rate is determined automatically. At start-up, and after a communication break, the initial rate is 9600 baud.
	(1) 9600	9600 baud.
	(2) 19200	19200 baud.
	(3) 38400	38400 baud.
	(4) 57600	57600 baud.
55.03	MDB PARITY	FW block: None
	Defines the use of parity bits. The same setting must be used in all on-line stations.	
	0…3	Number of parity bits.

Group 57 D2D COMMUNICATION

Drive-to-drive communication settings. See *Appendix B – Drive-to-drive link* on page 331.

Firmware block: **D2D COMMUNICATION** (57) This block sets up the drive-to-drive communication. It also shows the main drive-to-drive control word and the two references.		
Block outputs located in other parameter groups		2.17 D2D MAIN CW (page 66) 2.19 D2D REF1 (page 67) 2.20 D2D REF2 (page 67)
57.01	LINK MODE	FW block: D2D COMMUNICATION (see above)
	Activates the drive-to-drive connection.	
	(0) Disabled	Drive-to-drive connection disabled.
	(1) Follower	The drive is a follower on the drive-to-drive link.
	(2) Master	The drive is the master on the drive-to-drive link. Only one drive can be the master at a time.
57.02	COMM LOSS FUNC	FW block: D2D COMMUNICATION (see above)
	Selects how the drive acts when an erroneous drive-to-drive configuration or a communication break is detected.	
	(0) No	Protection inactive.
	(1) Alarm	The drive generates an alarm.
	(2) Fault	The drive trips on a fault.
57.03	NODE ADDRESS	FW block: D2D COMMUNICATION (see above)
	Sets the node address for a follower drive. Each follower must have a dedicated node address. **Note:** If the drive is set to be the master on the drive-to-drive link, this parameter has no effect (the master is automatically assigned node address 0).	

	1…62	Node address.
57.04	FOLLOWER MASK 1	FW block: D2D COMMUNICATION (see above)
	On the master drive, selects the followers to be polled. If no response is received from a polled follower, the action selected by parameter 57.02 COMM LOSS FUNC is taken. The least significant bit represents follower with node address 1, while the most significant bit represents follower 31. When a bit is set to 1, the corresponding node address is polled. For example, followers 1 and 2 are polled when this parameter is set to the value of 0x3.	
	0x00000000…0x7FFFFFFF	Follower mask 1.
57.05	FOLLOWER MASK 2	FW block: D2D COMMUNICATION (see above)
	On the master drive, selects the followers to be polled. If no response is received from a polled follower, the action selected by parameter 57.02 COMM LOSS FUNC is taken. The least significant bit represents follower with node address 32, while the most significant bit represents follower 62. When a bit is set to 1, the corresponding node address is polled. For example, followers 32 and 33 are polled when this parameter is set to the value of 0x3.	
	0x00000000…0x7FFFFFFF	Follower mask 2.
57.06	REF 1 SRC	FW block: D2D COMMUNICATION (see above)
	Selects the source of D2D reference 1 sent to the followers. The parameter is effective on the master drive, as well as submasters (57.03 NODE ADDRESS = 57.12 REF1 MC GROUP) in a multicast message chain (see parameter 57.11 REF 1 MSG TYPE). The default value is P.03.04, i.e. 3.04 SPEEDREF RAMPED.	
	Value pointer: Group and index.	
57.07	REF 2 SRC	FW block: D2D COMMUNICATION (see above)
	On the master drive, selects the source of D2D reference 2 broadcast to all followers. The default value is P.03.13, i.e. 3.13 TORQ REF TO TC.	
	Value pointer: Group and index.	
57.08	FOLLOWER CW SRC	FW block: D2D COMMUNICATION (see above)
	Selects the source of the D2D control word sent to the followers. The parameter is effective on the master drive, as well as submasters in a multicast message chain (see parameter 57.11 REF 1 MSG TYPE). The default value is P.02.18, i.e. 2.18 D2D FOLLOWER CW.	
	Value pointer: Group and index.	
57.09	KERNEL SYNC MODE	FW block: D2D COMMUNICATION (see above)
	Determines which signal the time levels of the drive are synchronised with. An offset can be defined by parameter 57.10 KERNEL SYNC OFFS if desired.	
	(0) NoSync	No synchronisation.
	(1) D2DSync	If the drive is the master on a drive-to-drive link, it broadcasts a synchronisation signal to the follower(s). If the drive is a follower, it synchronises its firmware time levels to the signal received from the master.

	(2) FBSync	The drive synchronises its firmware time levels to a synchronisation signal received through a fieldbus adapter.
	(3) FBToD2DSync	If the drive is the master on a drive-to-drive link, it synchronises its firmware time levels to a synchronisation signal received from a fieldbus adapter, and broadcasts the signal on the drive-to-drive link. If the drive is a follower, this setting has no effect.
57.10	KERNEL SYNC OFFS	FW block: D2D COMMUNICATION (see above)
	Defines an offset between the synchronisation signal received and the time levels of the drive. With a positive value, the drive time levels will lag behind the synchronisation signal; with a negative value, the drive time levels will lead.	
	-4999…5000 ms	Synchronisation offset.
57.11	REF 1 MSG TYPE	FW block: D2D COMMUNICATION (see above)
	By default, in drive-to-drive communication, the master broadcasts the drive-to-drive control word and references 1 and 2 to all followers. This parameter enables multicasting, i.e. sending the drive-to-drive control word and reference 1 to a certain drive or group of drives. The message can then be further relayed to another group of drives to form a multicast chain. In the master, as well as any submaster (i.e. follower relaying the message to other followers), the sources for the control word and reference 1 are selected by parameters 57.08 FOLLOWER CW SRC and 57.06 REF 1 SRC respectively. **Note:** Reference 2 is broadcast by the master to all followers. For more information, see *Appendix B – Drive-to-drive link* on page 331.	
	(0) Broadcast	The control word and reference 1 are sent by the master to all followers. If the master has this setting, the parameter has no effect in the followers.
	(1) Ref1 MC Grps	The drive-to-drive control word and reference 1 are only sent to the drives in the multicast group specified by parameter 57.13 NEXT REF1 MC GRP. This setting can also used in submasters (followers in which parameters 57.03 NODE ADDRESS and 57.12 REF1 MC GROUP are set to the same value) to form a multicast chain.
57.12	REF1 MC GROUP	FW block: D2D COMMUNICATION (see above)
	Selects the multicast group the drive belongs to. See parameter 57.11 REF 1 MSG TYPE.	
	0…62	Multicast group (0 = none).
57.13	NEXT REF1 MC GRP	FW block: D2D COMMUNICATION (see above)
	Specifies the next multicast group of drives the multicast message is relayed to. See parameter 57.11 REF 1 MSG TYPE. This parameter is effective only in the master or in submasters (followers in which parameters 57.03 NODE ADDRESS and 57.12 REF1 MC GROUP are set to the same value).	
	0…62	Next multicast group in message chain.

57.14	NR REF1 MC GRPS	FW block: D2D COMMUNICATION (see above)
	Sets the number of drives sending messages in the message chain. The value is typically equal to the number of multicast groups in the chain assuming that the last drive is NOT sending an acknowledgement to the master. See parameter 57.11 REF 1 MSG TYPE. **Notes:** • This parameter is only effective in the master.	
	1…62	Total number of links in multicast message chain.
57.15	D2D COMM PORT	FW block: None
	Defines the hardware to which the drive-to-drive link is connected. In special cases (such as harsh operating conditions), the galvanic isolation provided by the RS-485 interface of the FMBA module may make for more robust communication than the standard drive-to-drive connection.	
	(0) on-board	Connector X5 on the JCU Control Unit is used.
	(1) Slot 1	An FMBA module installed in JCU option slot 1 is used.
	(2) Slot 2	An FMBA module installed in JCU option slot 2 is used.
	(3) Slot 3	An FMBA module installed in JCU option slot 3 is used.

Group 90 ENC MODULE SEL

Settings for encoder activation, emulation, TTL echo, and encoder cable fault detection.

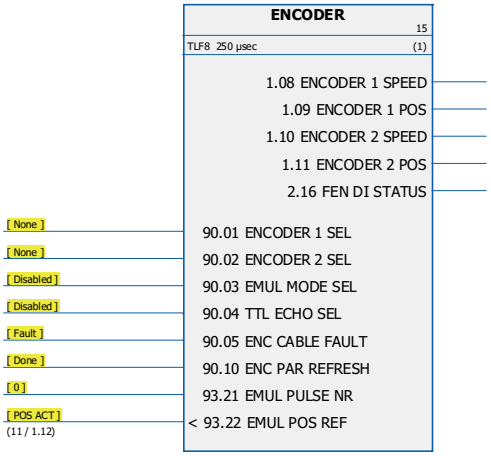
The firmware supports two encoders, encoder 1 and 2 (but only one FEN-21 Resolver Interface Module). Revolution counting is only supported for encoder 1. The following optional interface modules are available:

- TTL Encoder Interface Module FEN-01: two TTL inputs, TTL output (for encoder emulation and echo), two digital inputs for position latching, PTC temperature sensor connection
- Absolute Encoder Interface FEN-11: absolute encoder input, TTL input, TTL output (for encoder emulation and echo), two digital inputs for position latching, PTC/KTY temperature sensor connection
- Resolver Interface Module FEN-21: resolver input, TTL input, TTL output (for encoder emulation echo), two digital inputs for position latching, PTC/KTY temperature sensor connection
- HTL Encoder Interface Module FEN-31: HTL encoder input, TTL output (for encoder emulation and echo), two digital inputs for position latching, PTC/KTY temperature sensor connection.

The interface module is connected to drive option Slot 1 or 2. **Note:** Two encoder interface modules of the same type are not allowed.

For encoder/resolver configuration, see parameter groups 91 ABSOL ENC CONF (page 173), 92 RESOLVER CONF (page 178) and 93 PULSE ENC CONF (page 179).

Note: Configuration data is written into the logic registers of the interface module once after the power-up. If parameter values are changed, save values into the permanent memory using parameter 16.07 PARAM SAVE. The new settings will take effect when the drive is powered up again, or after re-configuration is forced using parameter 90.10 ENC PAR REFRESH.

Firmware block: **ENCODER** (3) This block • activates the communication to encoder interface 1/2 • enables encoder emulation/echo • shows encoder 1/2 speed and actual position.		
Block inputs located in other parameter groups		93.21 EMUL PULSE NR (page 181) 93.22 EMUL POS REF (page 181)
Block outputs located in other parameter groups		1.08 ENCODER 1 SPEED (page 59) 1.09 ENCODER 1 POS (page 59) 1.10 ENCODER 2 SPEED (page 60) 1.11 ENCODER 2 POS (page 60) 2.16 FEN DI STATUS (page 66)
90.01	ENCODER 1 SEL	FW block: ENCODER (see above)
	Activates the communication to optional encoder/resolver interface 1. **Note:** It is recommended that encoder interface 1 is used whenever possible since the data received through that interface is fresher than the data received through interface 2. On the other hand, when position values used in emulation are determined by the drive software, the use of encoder interface 2 is recommended as the values are transmitted earlier through interface 2 than through interface 1.	
	(0) None	Inactive.
	(1) FEN-01 TTL+	Communication active. Module type: FEN-01 TTL Encoder interface Module. Input: TTL encoder input with commutation support (X32). See parameter group 93 PULSE ENC CONF.
	(2) FEN-01 TTL	Communication active. Module type: FEN-01 TTL Encoder interface Module. Input: TTL encoder input (X31). See parameter group 93 PULSE ENC CONF.
	(3) FEN-11 ABS	Communication active. Module type: FEN-11 Absolute Encoder Interface. Input: Absolute encoder input (X42). See parameter group 91 ABSOL ENC CONF.
	(4) FEN-11 TTL	Communication active. Module type: FEN-11 Absolute Encoder Interface. Input: TTL encoder input (X41). See parameter group 93 PULSE ENC CONF.
	(5) FEN-21 RES	Communication active. Module type: FEN-21 Resolver Interface. Input: Resolver input (X52). See parameter group 92 RESOLVER CONF.

	(6) FEN-21 TTL	Communication active. Module type: FEN-21 Resolver Interface. Input: TTL encoder input (X51). See parameter group 93 PULSE ENC CONF.
	(7) FEN-31 HTL	Communication active. Module type: FEN-31 HTL Encoder Interface. Input: HTL encoder input (X82). See parameter group 93 PULSE ENC CONF.
90.02	ENCODER 2 SEL	FW block: ENCODER (see above)
	Activates the communication to the optional encoder/resolver interface 2. For selections, see parameter 90.01 ENCODER 1 SEL. **Note:** The counting of full shaft revolutions is not supported for encoder 2.	
90.03	EMUL MODE SEL	FW block: ENCODER (see above)
	Enables the encoder emulation and selects the position value and the TTL output used in the emulation process. In encoder emulation, a calculated position difference is transformed to a corresponding number of TTL pulses to be transmitted via the TTL output. The position difference is the difference between the latest and the previous position values. The position value used in emulation can be either a position determined by the drive software or a position measured by an encoder. If drive software position is used, the source for the used position is selected by parameter 93.22 EMUL POS REF. Because the software causes a delay, it is recommended that actual position is always taken from an encoder. Drive software is recommended to be used only with position reference emulation. Encoder emulation can be used to increase or decrease the pulse number when TTL encoder data is transmitted via the TTL output e.g. to another drive. If the pulse number requires no alternation, use encoder echo for data transformation. See parameter 90.04 TTL ECHO SEL. **Note:** If encoder emulation and echo are enabled for the same FEN-xx TTL output, the emulation overrides the echo. If an encoder input is selected as emulation source, the corresponding selection must be activated either with parameter 90.01 ENCODER 1 SEL or 90.02 ENCODER 2 SEL. The TTL encoder pulse number used in emulation must be defined by parameter 93.21 EMUL PULSE NR. See parameter group 93 PULSE ENC CONF.	
	(0) Disabled	Emulation disabled.
	(1) FEN-01 SWref	Module type: FEN-01 TTL Encoder interface Module. Emulation: Drive software position (source selected by par. 93.22 EMUL POS REF) is emulated to FEN-01 TTL output.
	(2) FEN-01 TTL+	Module type: FEN-01 TTL Encoder interface Module. Emulation: FEN-01 TTL encoder input (X32) position is emulated to FEN-01 TTL output.
	(3) FEN-01 TTL	Module type: FEN-01 TTL Encoder interface Module. Emulation: FEN-01 TTL encoder input (X31) position is emulated to FEN-01 TTL output.
	(4) FEN-11 SWref	Module type: FEN-11 Absolute Encoder Interface. Emulation: Drive software position (source selected by par. 93.22 EMUL POS REF) is emulated to FEN-11 TTL output.
	(5) FEN-11 ABS	Module type: FEN-11 Absolute Encoder Interface. Emulation: FEN-11 absolute encoder input (X42) position is emulated to FEN-11 TTL output.

	(6) FEN-11 TTL	Module type: FEN-11 Absolute Encoder Interface. Emulation: FEN-11 TTL encoder input (X41) position is emulated to FEN-11 TTL output.
	(7) FEN-21 SWref	Module type: FEN-21 Resolver Interface. Emulation: Drive software position (source selected by par. 93.22 EMUL POS REF) is emulated to FEN-21 TTL output.
	(8) FEN-21 RES	Module type: FEN-21 Resolver Interface. Emulation: FEN-21 resolver input (X52) position is emulated to FEN-21 TTL output.
	(9) FEN-21 TTL	Module type: FEN-21 Resolver Interface. Emulation: FEN-21 TTL encoder input (X51) position is emulated to FEN-21 TTL output.
	(10) FEN-31 SWref	Module type: FEN-31 HTL Encoder Interface. Emulation: Drive software position (source selected by par. 93.22 EMUL POS REF) is emulated to FEN-31 TTL output.
	(11) FEN-31 HTL	Module type: FEN-31 HTL Encoder Interface. Emulation: FEN-31 HTL encoder input (X82) position is emulated to FEN-31 TTL output.
90.04	TTL ECHO SEL	FW block: ENCODER (see above)
	Enables and selects the interface for the TTL encoder signal echo. **Note:** If encoder emulation and echo are enabled for the same FEN-xx TTL output, the emulation overrides the echo.	
	(0) Disabled	TTL echo disabled.
	(1) FEN-01 TTL+	Module type: FEN-01 TTL Encoder Interface. Echo: TTL encoder input (X32) pulses are echoed to the TTL output.
	(2) FEN-01 TTL	Module type: FEN-01 TTL Encoder Interface. Echo: TTL encoder input (X31) pulses are echoed to the TTL output.
	(3) FEN-11 TTL	Module type: FEN-11 Absolute Encoder Interface. Echo: TTL encoder input (X41) pulses are echoed to the TTL output.
	(4) FEN-21 TTL	Module type: FEN-21 Resolver Interface. Echo: TTL encoder input (X51) pulses are echoed to the TTL output.
	(5) FEN-31 HTL	Module type: FEN-31 HTL Encoder Interface. Echo: HTL encoder input (X82) pulses are echoed to the TTL output.
90.05	ENC CABLE FAULT	FW block: ENCODER (see above)
	Selects the action in case an encoder cable fault is detected by the FEN-xx encoder interface. **Notes:** • At the time of printing, this functionality is only available with the absolute encoder input of the FEN-11 based on sine/cosine incremental signals, and with the HTL input of the FEN-31. • When the encoder input is used for speed feedback (see 22.01 SPEED FB SEL), this parameter may be overridden by parameter 22.09 SPEED FB FAULT.	
	(0) No	Cable fault detection inactive.
	(1) Fault	The drive trips on an ENCODER 1/2 CABLE fault.

	(2) Warning	The drive generates an ENCODER 1/2 CABLE warning. This is the recommended setting if the maximum pulse frequency of sine/cosine incremental signals exceeds 100 kHz; at high frequencies, the signals may attenuate enough to invoke the function. The maximum pulse frequency can be calculated as follows: $\frac{\text{Pulses per revolution (par. 91.01)} \times \text{Maximum speed in rpm}}{60}$
90.10	ENC PAR REFRESH	FW block: ENCODER (see above)
	Setting this parameter to 1 forces a reconfiguration of the FEN-xx interfaces, which is needed for any parameter changes in groups 90…93 to take effect. **Note:** This parameter cannot be changed while the drive is running.	
	(0) Done	Refreshing done.
	(1) Configure	Reconfigure. The value will automatically revert to DONE.

Group 91 ABSOL ENC CONF

Absolute encoder configuration; used when parameter 90.01 ENCODER 1 SEL / 90.02 ENCODER 2 SEL is set to (3) FEN-11 ABS.

The optional FEN-11 Absolute Encoder Interface module supports the following encoders:

- Incremental sin/cos encoders with or without zero pulse and with or without sin/cos commutation signals
- Endat 2.1/2.2 with incremental sin/cos signals (partially without sin/cos incremental signals*)
- Hiperface encoders with incremental sin/cos signals
- SSI (Synchronous Serial Interface) with incremental sin/cos signals (partially without sin/cos incremental signals*)
- Tamagawa 17/33-bit digital encoders (the resolution of position data within one revolution is 17 bits; multiturn data includes a 16-bit revolution count).

* EnDat and SSI encoders without incremental sin/cos signals are partially supported only as encoder 1: Speed is not available and the time instant of the position data (delay) depends on the encoder.

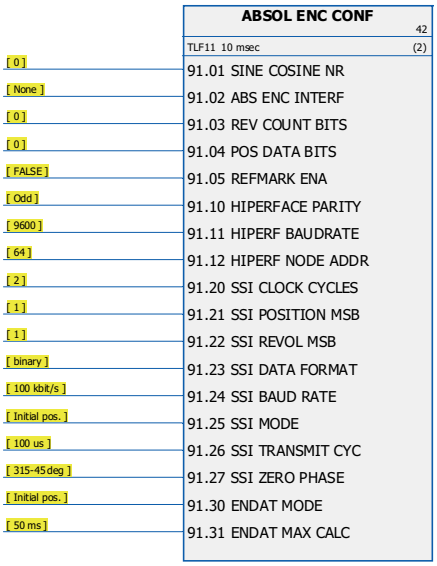
See also parameter group 90 ENC MODULE SEL on page 169, and *FEN-11 Absolute Encoder Interface User's Manual* (3AFE68784841 [English]).

Note: Configuration data is written into the logic registers of the interface module once after the power-up. If parameter values are changed, save values into the permanent memory using parameter 16.07 PARAM SAVE. The new settings will take effect when the drive is powered up again, or after re-configuration is forced using parameter 90.10 ENC PAR REFRESH.

Firmware block:

ABSOL ENC CONF
(91)

This block configures the absolute encoder connection.



91.01	SINE COSINE NR	FW block: ABSOL ENC CONF (see above)
	Defines the number of sine/cosine wave cycles within one revolution. **Note:** This parameter does not need to be set when EnDat or SSI encoders are used in continuous mode. See parameter 91.25 SSI MODE / 91.30 ENDAT MODE.	
	0…65535	Number of sine/cosine wave cycles within one revolution.
91.02	ABS ENC INTERF	FW block: ABSOL ENC CONF (see above)
	Selects the source for the encoder absolute position.	
	(0) None	Not selected.
	(1) Commut sig	Commutation signals.
	(2) EnDat	Serial interface: EnDat encoder.
	(3) Hiperface	Serial interface: HIPERFACE encoder.
	(4) SSI	Serial interface: SSI encoder.
	(5) Tamag. 17/33B	Serial interface: Tamagawa 17/33-bit encoder.
91.03	REV COUNT BITS	FW block: ABSOL ENC CONF (see above)
	Defines the number of bits used in revolution counting with multiturn encoders. Used when parameter 91.02 ABS ENC INTERF is set to (2) EnDat, (3) Hiperface or (4) SSI. When 91.02 ABS ENC INTERF is set to (5) Tamag. 17/33B), setting this parameter to a non-zero value activates multiturn data requesting.	
	0…32	Number of bits used in revolution count. E.g. 4096 revolutions => 12 bits.
91.04	POS DATA BITS	FW block: ABSOL ENC CONF (see above)
	Defines the number of bits used within one revolution when parameter 91.02 ABS ENC INTERF is set to (2) EnDat, (3) Hiperface or (4) SSI. When 91.02 ABS ENC INTERF is set to (5) Tamag. 17/33B, this parameter is internally set to 17.	
	0…32	Number of bits used within one revolution. E.g. 32768 positions per revolution => 15 bits.
91.05	REFMARK ENA	FW block: ABSOL ENC CONF (see above)
	Enables the encoder zero pulse for the absolute encoder input (X42) of an FEN-11 module (if present). Zero pulse can be used for position latching. **Note:** With serial interfaces (i.e. when parameter 91.02 ABS ENC INTERF is set to (2) EnDat, (3) Hiperface, (4) SSI or (5) Tamag. 17/33B), the zero pulse does not exist.	
	(0) FALSE	Zero pulse disabled.
	(1) TRUE	Zero pulse enabled.
91.10	HIPERFACE PARITY	FW block: ABSOL ENC CONF (see above)
	Defines the use of parity and stop bit(s) for HIPERFACE encoder (i.e. when parameter 91.02 ABS ENC INTERF is set to (3) Hiperface). Typically this parameter does not need to be set.	

	(0) Odd	Odd parity indication bit, one stop bit.
	(1) Even	Even parity indication bit, one stop bit.
91.11	HIPERF BAUDRATE	FW block: ABSOL ENC CONF (see above)
	Defines the transfer rate of the link for HIPERFACE encoder (i.e. when parameter 91.02 ABS ENC INTERF is set to (3) Hiperface). Typically this parameter does not need to be set.	
	(0) 4800	4800 bits/s.
	(1) 9600	9600 bits/s.
	(2) 19200	19200 bits/s.
	(3) 38400	38400 bits/s.
91.12	HIPERF NODE ADDR	FW block: ABSOL ENC CONF (see above)
	Defines the node address for HIPERFACE encoder (i.e. when parameter 91.02 ABS ENC INTERF is set to (3) Hiperface). Typically this parameter does not need to be set.	
	0…255	HIPERFACE encoder node address.
91.20	SSI CLOCK CYCLES	FW block: ABSOL ENC CONF (see above)
	Defines the length of the SSI message. The length is defined as the number of clock cycles. The number of cycles can be calculated by adding 1 to the number of the bits in a SSI message frame. Used with SSI encoders, i.e. when parameter 91.02 ABS ENC INTERF is set to (4) SSI.	
	2…127	SSI message length.
91.21	SSI POSITION MSB	FW block: ABSOL ENC CONF (see above)
	Defines the location of the MSB (main significant bit) of the position data within a SSI message. Used with SSI encoders, i.e. when parameter 91.02 ABS ENC INTERF is set to (4) SSI.	
	1…126	Position data MSB location (bit number).
91.22	SSI REVOL MSB	FW block: ABSOL ENC CONF (see above)
	Defines the location of the MSB (main significant bit) of the revolution count within a SSI message. Used with SSI encoders, i.e. when parameter 91.02 ABS ENC INTERF is set to (4) SSI.	
	1…126	Revolution count MSB location (bit number).
91.23	SSI DATA FORMAT	FW block: ABSOL ENC CONF (see above)
	Selects the data format for SSI encoder (i.e. when parameter 91.02 ABS ENC INTERF is set to (4) SSI).	
	(0) binary	Binary code.
	(1) gray	Gray code.

91.24	SSI BAUD RATE	FW block: ABSOL ENC CONF (see above)
	Selects the baud rate for SSI encoder (i.e. when parameter 91.02 ABS ENC INTERF is set to (4) SSI).	
	(0) 10 kbit/s	10 kbit/s.
	(1) 50 kbit/s	50 kbit/s.
	(2) 100 kbit/s	100 kbit/s.
	(3) 200 kbit/s	200 kbit/s.
	(4) 500 kbit/s	500 kbit/s.
	(5) 1000 kbit/s	1000 kbit/s.
91.25	SSI MODE	FW block: ABSOL ENC CONF (see above)
	Selects the SSI encoder mode. **Note:** Parameter needs to be set only when an SSI encoder is used in continuous mode, i.e. SSI encoder without incremental sin/cos signals (supported only as encoder 1). SSI encoder is selected by setting parameter 91.02 ABS ENC INTERF to (4) SSI.	
	(0) Initial pos.	Single position transfer mode (initial position).
	(1) Continuous	Continuous position transfer mode.
91.26	SSI TRANSMIT CYC	FW block: ABSOL ENC CONF (see above)
	Selects the transmission cycle for SSI encoder. **Note:** This parameter needs to be set only when an SSI encoder is used in continuous mode, i.e. SSI encoder without incremental sin/cos signals (supported only as encoder 1). SSI encoder is selected by setting parameter 91.02 ABS ENC INTERF to (4) SSI.	
	(0) 50 us	50 µs.
	(1) 100 us	100 µs.
	(2) 200 us	200 µs.
	(3) 500 us	500 µs.
	(4) 1 ms	1 ms.
	(5) 2 ms	2 ms.
91.27	SSI ZERO PHASE	FW block: ABSOL ENC CONF (see above)
	Defines the phase angle within one sine/cosine signal period that corresponds to the value of zero on the SSI serial link data. The parameter is used to adjust the synchronization of the SSI position data and the position based on sine/cosine incremental signals. Incorrect synchronization may cause an error of ±1 incremental period. **Note:** This parameter needs to be set only when an SSI encoder with sine/cosine incremental signals is used in initial position mode.	
	(0) 315–45 deg	315–45 degrees.
	(1) 45–135 deg	45–135 degrees.

	(2) 135–225 deg	135–225 degrees.
	(3) 225–315 deg	225–315 degrees.
91.30	ENDAT MODE	FW block: ABSOL ENC CONF (see above)
	Selects the EnDat encoder mode. **Note:** This parameter needs to be set only when an EnDat encoder is used in continuous mode, i.e. EnDat encoder without incremental sin/cos signals (supported only as encoder 1). EnDat encoder is selected by setting parameter 91.02 ABS ENC INTERF to (2) EnDat.	
	(0) Initial pos.	Single position transfer mode (initial position).
	(1) Continuous	Continuous position data transfer mode.
91.31	ENDAT MAX CALC	FW block: ABSOL ENC CONF (see above)
	Selects the maximum encoder calculation time for EnDat encoder. **Note:** This parameter needs to be set only when an EnDat encoder is used in continuous mode, i.e. EnDat encoder without incremental sin/cos signals (supported only as encoder 1). EnDat encoder is selected by setting parameter 91.02 ABS ENC INTERF to (2) EnDat.	
	(0) 10 us	10 µs.
	(1) 100 us	100 µs.
	(2) 1 ms	1 ms.
	(3) 50 ms	50 ms.

Group 92 RESOLVER CONF

Resolver configuration; used when parameter 90.01 ENCODER 1 SEL / 90.02 ENCODER 2 SEL is set to (5) FEN-21 RES.

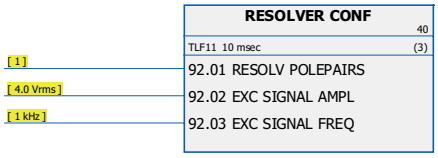
The optional FEN-21 Resolver Interface module is compatible with resolvers which are excited by sinusoidal voltage (to the rotor winding) and which generate sine and cosine signals proportional to the rotor angle (to stator windings).

Note: Configuration data is written into the logic registers of the adapter once after the power-up. If parameter values are changed, save values into the permanent memory by parameter 16.07 PARAM SAVE. The new settings will take effect when the drive is powered up again, or after re-configuration is forced by parameter 90.10 ENC PAR REFRESH.

Resolver autotuning is performed automatically whenever the resolver input is activated after changes to parameters 92.02 EXC SIGNAL AMPL or 92.03 EXC SIGNAL FREQ. Autotuning must be forced after any changes in the resolver cable connection. This can be done by setting either 92.02 EXC SIGNAL AMPL or 92.03 EXC SIGNAL FREQ to its already existing value, and then setting parameter 90.10 ENC PAR REFRESH to 1.

If the resolver (or absolute encoder) is used for feedback from a permanent magnet motor, an AUTOPHASING ID run should be performed after replacement or any parameter changes. See parameter 99.13 IDRUN MODE and section *Autophasing* on page 38.

See also parameter group 90 ENC MODULE SEL on page 169, and *FEN-21 Resolver Interface User's Manual* (3AFE68784859 [English]).

Firmware block: **RESOLVER CONF** (92) This block configures the resolver connection.		
92.01	RESOLV POLEPAIRS	FW block: RESOLVER CONF (see above)
	Selects the number of pole pairs.	
	1…32	Number of pole pairs.
92.02	EXC SIGNAL AMPL	FW block: RESOLVER CONF (see above)
	Defines the amplitude of the excitation signal.	
	4.0…12.0 Vrms	Excitation signal amplitude.
92.03	EXC SIGNAL FREQ	FW block: RESOLVER CONF (see above)
	Defines the frequency of the excitation signal.	
	1…20 kHz	Excitation signal frequency.

Group 93 PULSE ENC CONF

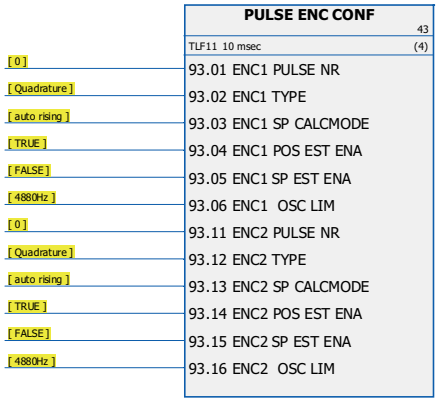
TTL/HTL input and TTL output configuration. See also parameter group 90 ENC MODULE SEL on page 169, and the appropriate encoder extension module manual.

Parameters 93.01…93.06 are used when a TTL/HTL encoder is used as encoder 1 (see parameter 90.01 ENCODER 1 SEL).

Parameters 93.11…93.16 are used when a TTL/HTL encoder is used as encoder 2 (see parameter 90.02 ENCODER 2 SEL).

Typically, only parameter 93.01/93.11 needs to be set for TTL/HTL encoders.

Note: Configuration data is written into the logic registers of the adapter once after the power-up. If parameter values are changed, save values into the permanent memory by parameter 16.07 PARAM SAVE. The new settings will take effect when the drive is powered up again, or after re-configuration is forced by parameter 90.10 ENC PAR REFRESH.

Firmware block: **PULSE ENC CONF** (93) This block configures the TTL/HTL input and TTL output.		
93.01	ENC1 PULSE NR	FW block: PULSE ENC CONF (see above)
	Defines the pulse number per revolution for encoder 1.	
	0…65535	Pulses per revolution for encoder 1.
93.02	ENC1 TYPE	FW block: PULSE ENC CONF (see above)
	Selects the type of encoder 1.	
	(0) Quadrature	Quadrature encoder (two channels, channels A and B).
	(1) single track	Single track encoder (one channel, channel A).
93.03	ENC1 SP CALCMODE	FW block: PULSE ENC CONF (see above)
	Selects the speed calculation mode for encoder 1. *When single track mode has been selected by parameter 93.02 ENC1 TYPE, the speed is always positive.	

<table>
<tr><td></td><td>(0) A&B all</td><td>Channels A and B: Rising and falling edges are used for speed calculation. Channel B: Defines the direction of rotation. *
Note: When single track mode has been selected by parameter 93.02 ENC1 TYPE, setting 0 acts like setting 1.</td></tr>
<tr><td></td><td>(1) A all</td><td>Channel A: Rising and falling edges are used for speed calculation. Channel B: Defines the direction of rotation. *</td></tr>
<tr><td></td><td>(2) A rising</td><td>Channel A: Rising edges are used for speed calculation. Channel B: Defines the direction of rotation. *</td></tr>
<tr><td></td><td>(3) A falling</td><td>Channel A: Falling edges are used for speed calculation. Channel B: Defines the direction of rotation. *</td></tr>
<tr><td></td><td>(4) auto rising

(5) auto falling</td><td>Used mode (0, 1, 2 or 3) is changed automatically depending on the pulse frequency according to the following table:
<table>
<tr><th>93.03 = 4</th><th>93.03 = 5</th><th rowspan="2">Pulse frequency of the channel(s)</th></tr>
<tr><th colspan="2">Used mode</th></tr>
<tr><td>0</td><td>0</td><td>< 2442 Hz</td></tr>
<tr><td>1</td><td>1</td><td>2442…4884 Hz</td></tr>
<tr><td>2</td><td>3</td><td>> 4884 Hz</td></tr>
</table></td></tr>
<tr><td>93.04</td><td>ENC1 POS EST ENA</td><td>FW block: PULSE ENC CONF (see above)</td></tr>
<tr><td></td><td colspan="2">Selects whether position estimation is used with encoder 1 to increase position data resolution or not.</td></tr>
<tr><td></td><td>(0) FALSE</td><td>Measured position (Resolution: 4 x pulses per revolution for quadrature encoders, 2 x pulses per revolution for single track encoders.)</td></tr>
<tr><td></td><td>(1) TRUE</td><td>Estimated position. (Uses position extrapolation. Extrapolated at the time of data request.)</td></tr>
<tr><td>93.05</td><td>ENC1 SP EST ENA</td><td>FW block: PULSE ENC CONF (see above)</td></tr>
<tr><td></td><td colspan="2">Selects whether calculated or estimated speed is used with encoder 1.</td></tr>
<tr><td></td><td>(0) FALSE</td><td>Last calculated speed (calculation interval is 62.5 µs…4 ms).</td></tr>
<tr><td></td><td>(1) TRUE</td><td>Estimated speed (estimated at the time of data request) Estimation increases the speed ripple in steady state operation, but improves the dynamics.</td></tr>
<tr><td>93.06</td><td>ENC1 OSC LIM</td><td>FW block: PULSE ENC CONF (see above)</td></tr>
<tr><td></td><td colspan="2">Activates transient filter for encoder 1. Changes of direction of rotation are ignored above the selected pulse frequency.</td></tr>
<tr><td></td><td>(0) 4880Hz</td><td>Change in rotation of direction allowed below 4880 Hz.</td></tr>
<tr><td></td><td>(1) 2440Hz</td><td>Change in rotation of direction allowed below 2440 Hz.</td></tr>
<tr><td></td><td>(2) 1220Hz</td><td>Change in rotation of direction allowed below 1220 Hz.</td></tr>
<tr><td></td><td>(3) Disabled</td><td>Change in rotation of direction allowed at any pulse frequency.</td></tr>
</table>

93.11	ENC2 PULSE NR	FW block: PULSE ENC CONF (see above)
	Defines the pulse number per revolution for encoder 2.	
	0…65535	Pulses per revolution for encoder 2.
93.12	ENC2 TYPE	FW block: PULSE ENC CONF (see above)
	Selects the type of encoder 2. For selections, see parameter 93.02 ENC1 TYPE.	
93.13	ENC2 SP CALCMODE	FW block: PULSE ENC CONF (see above)
	Selects the speed calculation mode for encoder 2. For selections, see parameter 93.03 ENC1 SP CALCMODE.	
93.14	ENC2 POS EST ENA	FW block: PULSE ENC CONF (see above)
	Selects whether measured or estimated position is used with encoder 2. For selections, see parameter 93.04 ENC1 POS EST ENA.	
93.15	ENC2 SP EST ENA	FW block: PULSE ENC CONF (see above)
	Selects whether calculated or estimated speed is used with encoder 2. For selections, see parameter 93.05 ENC1 SP EST ENA.	
93.16	ENC2 OSC LIM	FW block: PULSE ENC CONF (see above)
	Activates transient filter for encoder 2. Changes of direction of rotation are ignored above the selected pulse frequency. For selections, see parameter 93.06 ENC1 OSC LIM.	
93.21	EMUL PULSE NR	FW block: ENCODER (page 169)
	Defines the number of TTL pulses per revolution used in encoder emulation. Encoder emulation is enabled by parameter 90.03 EMUL MODE SEL.	
	0…65535	TTL pulses used in encoder emulation.
93.22	EMUL POS REF	FW block: ENCODER (page 169)
	Selects the source for the position value used in encoder emulation when parameter 90.03 EMUL MODE SEL is set to (1) FEN-01 SWref, (4) FEN-11 SWref, (7) FEN-21 SWref or (10) FEN-31 SWref. See parameter group 90 ENC MODULE SEL. The source can be any actual or reference position value (except 1.09 ENCODER 1 POS and 1.11 ENCODER 2 POS).	
	Value pointer: Group and index	
93.23	EMUL POS OFFSET	FW block: None
	Defines the zero point for emulated position in relation of the zero point of the input position (within one revolution). The input position is selected by parameter 90.03 EMUL MODE SEL. For example, if the offset is 0, an emulated zero pulse is generated each time the input position moves across 0. With an offset of 0.5, an emulated zero pulse is generated each time the input position (within one revolution) moves across 0.5.	
	0 … 0.99998 rev	Emulated zero pulse position offset.

Group 95 HW CONFIGURATION

Miscellaneous hardware-related settings.

95.01	CTRL UNIT SUPPLY	FW block: None
	Defines the manner in which the drive control unit is powered.	
	(0) Internal 24V	The drive control unit is powered from the drive power unit it is mounted on.
	(1) External 24V	The drive control unit is powered from an external power supply.
95.02	EXTERNAL CHOKE	FW block: None
	Defines if the drive is equipped with an AC choke or not.	
	(0) NO	The drive is not equipped with an AC choke.
	(1) YES	The drive is equipped with an AC choke.

Group 97 USER MOTOR PAR

User adjustment of motor model values estimated during ID run. The values can be entered in either “per unit” or SI.

97.01	USE GIVEN PARAMS	FW block: None
	Activates the motor model parameters 97.02…97.14 and the rotor angle offset parameter 97.20. **Notes:** • Parameter value is automatically set to zero when ID run is selected by parameter 99.13 IDRUN MODE. The values of parameters 97.02…97.20 are updated according to the motor characteristics identified during the ID run. • This parameter cannot be changed while the drive is running.	
	(0) NoUserPars	Parameters 97.02…97.20 inactive.
	(1) UserMotPars	The values of parameters 97.02…97.14 are used in the motor model.
	(2) UserPosOffs	The value of parameter 97.20 is used as the rotor angle offset. Parameters 97.02…97.14 are inactive.
	(3) AllUserPars	The values of parameters 97.02…97.14 are used in the motor model, and the value of parameter 97.20 is used as the rotor angle offset.
97.02	RS USER	FW block: None
	Defines the stator resistance R_S of the motor model.	
	0…0.5 p.u. (per unit)	Stator resistance.
97.03	RR USER	FW block: None
	Defines the rotor resistance R_R of the motor model. **Note:** This parameter is valid only for asynchronous motors.	
	0…0.5 p.u. (per unit)	Rotor resistance.
97.04	LM USER	FW block: None
	Defines the main inductance L_M of the motor model. **Note:** This parameter is valid only for asynchronous motors.	
	0…10 p.u. (per unit)	Main inductance.
97.05	SIGMAL USER	FW block: None
	Defines the leakage inductance σL_S. **Note:** This parameter is valid only for asynchronous motors.	
	0…1 p.u. (per unit)	Leakage inductance.
97.06	LD USER	FW block: None
	Defines the direct axis (synchronous) inductance. **Note:** This parameter is valid only for permanent magnet motors.	
	0…10 p.u. (per unit)	Direct axis (synchronous) inductance.

97.07	LQ USER	FW block: None
	Defines the quadrature axis (synchronous) inductance. **Note:** This parameter is valid only for permanent magnet motors.	
	0…10 p.u. (per unit)	Quadrature axis (synchronous) inductance.
97.08	PM FLUX USER	FW block: None
	Defines the permanent magnet flux. **Note:** This parameter is valid only for permanent magnet motors.	
	0…2 p.u. (per unit)	Permanent magnet flux.
97.09	RS USER SI	FW block: None
	Defines the stator resistance R_S of the motor model.	
	0.00000…100.00000 ohm	Stator resistance.
97.10	RR USER SI	FW block: None
	Defines the rotor resistance R_R of the motor model. **Note:** This parameter is valid only for asynchronous motors.	
	0.00000…100.00000 ohm	Rotor resistance.
97.11	LM USER SI	FW block: None
	Defines the main inductance L_M of the motor model. **Note:** This parameter is valid only for asynchronous motors.	
	0.00…100000.00 mH	Main inductance.
97.12	SIGL USER SI	FW block: None
	Defines the leakage inductance σL_S. **Note:** This parameter is valid only for asynchronous motors.	
	0.00…100000.00 mH	Leakage inductance.
97.13	LD USER SI	FW block: None
	Defines the direct axis (synchronous) inductance. **Note:** This parameter is valid only for permanent magnet motors.	
	0.00…100000.00 mH	Direct axis (synchronous) inductance.
97.14	LQ USER SI	FW block: None
	Defines the quadrature axis (synchronous) inductance. **Note:** This parameter is valid only for permanent magnet motors.	
	0.00…100000.00 mH	Quadrature axis (synchronous) inductance.

97.20	POS OFFSET USER	FW block: None
	Defines an angle offset between the zero position of the synchronous motor and the zero position of the position sensor. **Notes:** • The value is in electrical degrees. The electrical angle equals the mechanical angle multiplied by the number of motor pole pairs. • This parameter is valid only for permanent magnet motors.	
	0…360°	Angle offset.

Group 98 MOTOR CALC VALUES

Calculated motor values.

98.01	TORQ NOM SCALE	FW block: None
	Nominal torque in N•m which corresponds to 100%. **Note:** This parameter is copied from parameter 99.12 MOT NOM TORQUE if given. Otherwise the value is calculated.	
	0…2147483 Nm	Nominal torque.
98.02	POLEPAIRS	FW block: None
	Calculated number of motor pole pairs. **Note:** This parameter cannot be set by the user.	
	0…1000	Calculated number of motor pole pairs.

Group 99 START-UP DATA

Start-up settings such as language, motor data and motor control mode.

The nominal motor values must be set before the drive is started; for detailed instructions, see chapter *Start-up* on page 15.

With DTC motor control mode, parameters 99.06…99.10 must be set; better control accuracy is achieved by also setting parameters 99.11 and 99.12.

With scalar control, parameters 99.06…99.09 must be set.

99.01	LANGUAGE	FW block: None
	Selects the language. **Note:** Not all languages listed below are necessarily supported.	
	(0809h) ENGLISH	English.
	(0407h) DEUTSCH	German.
	(0410h) ITALIANO	Italian.
	(040Ah) ESPAÑOL	Spanish.
	(041Dh) SVENSKA	Swedish.
	(041Fh) TÜRKÇE	Turkish.
	(0419h) RUSSKI	Russian.
99.04	MOTOR TYPE	FW block: None
	Selects the motor type. **Note:** This parameter cannot be changed while the drive is running.	
	(0) AM	Asynchronous motor. Three phase AC voltage supplied induction motor with squirrel cage rotor.
	(1) PMSM	Permanent magnet motor. Three phase AC voltage supplied synchronous motor with permanent magnet rotor and sinusoidal BackEMF voltage.

99.05	MOTOR CTRL MODE	FW block: None
	Selects the motor control mode. DTC (Direct torque control) mode is suitable for most applications. Scalar control is suitable for special cases where DTC cannot be applied. In Scalar Control, the drive is controlled with a frequency reference. The outstanding motor control accuracy of DTC cannot be achieved in scalar control. There are some standard features that are disabled in the scalar control mode, for example motor identification run (99.13), torque limits in parameter group 20 LIMITS, DC hold and DC magnetising (11.04…11.06, 11.01). **Note:** Correct motor run requires that the magnetising current of the motor does not exceed 90 percent of the nominal current of the inverter. **Note:** Scalar control mode must be used • with multimotor applications 1) if the load is not equally shared between the motors, 2) if the motors are of different sizes, or 3) if the motors are going to be changed after the motor identification, • if the nominal current of the motor is less than 1/6 of the nominal output current of the drive, or • if the drive is used with no motor connected (e.g. for test purposes).	
	(0) DTC	Direct torque control mode.
	(1) Scalar	Scalar control mode.
99.06	MOT NOM CURRENT	FW block: None
	Defines the nominal motor current. Must be equal to the value on the motor rating plate. If several motors are connected to the inverter, enter the total current of the motors. **Note:** Correct motor run requires that the magnetising current of the motor does not exceed 90 percent of the nominal current of the inverter. **Note:** This parameter cannot be changed while the drive is running.	
	0…32767 A	Nominal motor current. **Note:** The allowed range is 1/6…2 × I_{2N} of drive for direct control mode (parameter 99.05 MOTOR CTRL MODE = (0) DTC). For scalar control mode (parameter 99.05 MOTOR CTRL MODE = (1) Scalar), the allowed range is 0…2 × I_{2N} of drive.
99.07	MOT NOM VOLTAGE	FW block: None
	Defines the nominal motor voltage. Nominal voltage is a fundamental phase to phase rms voltage, which is supplied to the motor at the nominal operating point. This parameter value must be equal to the value on the asynchronous motor name plate. **Note:** Make sure the motor is connected correctly (star or delta) in accordance to the rating plate. **Note:** With permanent magnet motors, the nominal voltage is the BackEMF voltage (at motor nominal speed). If the voltage is given as voltage per rpm, e.g. 60 V per 1000 rpm, the voltage for 3000 rpm nominal speed is 3 × 60 V = 180 V. Note that the nominal voltage is not equal to the equivalent DC motor voltage (E.D.C.M.) value given by some motor manufactures. The nominal voltage can be calculated by dividing the E.D.C.M. voltage by 1.7 (= square root of 3). **Note:** The stress on the motor insulations is always dependent on the drive supply voltage. This also applies to the case where the motor voltage rating is lower than the rating of the drive and the supply of the drive. **Note:** This parameter cannot be changed while the drive is running.	
	0…32767 V	Nominal motor voltage. **Note:** The allowed range is 1/6…2 × U_N of drive.

99.08	MOT NOM FREQ	FW block: None
	Defines the nominal motor frequency. **Note:** This parameter cannot be changed while the drive is running.	
	5…500 Hz	Nominal motor frequency.
99.09	MOT NOM SPEED	FW block: None
	Defines the nominal motor speed. Must be equal to the value on the motor rating plate. When parameter value is changed, check the speed limits in parameter group 20 LIMITS. **Note:** This parameter cannot be changed while the drive is running.	
	0…30000 rpm	Nominal motor speed.
99.10	MOT NOM POWER	FW block: None
	Defines the nominal motor power. Must be equal to the value on the motor rating plate. If several motors are connected to the inverter, enter the total power of the motors. Set also parameter 99.11 MOT NOM COSFII. **Note:** This parameter cannot be changed while the drive is running.	
	0…10000 kW	Nominal motor power.
99.11	MOT NOM COSFII	FW block: None
	Defines the cosphi (not applicable to permanent magnet motors) for a more accurate motor model. Not obligatory; if set, should be equal to the value on the motor rating plate. **Note:** This parameter cannot be changed while the drive is running.	
	0…1	Cosphi (0 = parameter disabled).
99.12	MOT NOM TORQUE	FW block: None
	Defines the nominal motor shaft torque for a more accurate motor model. Not obligatory. **Note:** This parameter cannot be changed while the drive is running.	
	0…2147483 Nm	Nominal motor shaft torque.

99.13	IDRUN MODE	FW block: None
	Selects the type of the motor identification performed at the next start of the drive in DTC mode. During the identification, the drive will identify the characteristics of the motor for optimum motor control. After the ID run, the drive is stopped. **Note:** This parameter cannot be changed while the drive is running. Once the ID run is activated, it can be cancelled by stopping the drive: If ID run has already been performed once, parameter is automatically set to (0) No. If no ID run has been performed yet, parameter is automatically set to (3) Standstill. In this case, the ID run must be performed. **Notes:** • ID run can only be performed in local control (i.e. when drive is controlled via PC tool or control panel). • ID run cannot be performed if parameter 99.05 MOTOR CTRL MODE is set to (1) Scalar. • ID run must be performed every time any of the motor parameters (99.04, 99.06…99.12) have been changed. Parameter is automatically set to STANDSTILL after the motor parameters have been set. • With permanent magnet motor, the motor shaft must NOT be locked and the load torque must be < 10% during the ID run (Normal/Reduced/Standstill). • Mechanical brake (if present) is not opened during the ID run. • Ensure that possible Safe Torque Off and emergency stop circuits are closed during ID run.	
	(0) No	No motor ID run is requested. This mode can be selected only if the ID run (Normal/Reduced/Standstill) has already been performed once.
	(1) Normal	Guarantees the best possible control accuracy. The ID run takes about 90 seconds. This mode should be selected whenever it is possible. **Note:** The driven machinery must be de-coupled from the motor with Normal ID run: • if the load torque is higher than 20%. • if the machinery is not able to withstand the nominal torque transient during the ID run. **Note:** Check the direction of rotation of the motor before starting the ID run. During the run, the motor will rotate in the forward direction. **WARNING!** The motor will run at up to approximately 50…100% of the nominal speed during the ID run. ENSURE THAT IT IS SAFE TO RUN THE MOTOR BEFORE PERFORMING THE ID RUN!
	(2) Reduced	Reduced ID Run. This mode should be selected instead of the Normal ID Run • if mechanical losses are higher than 20% (i.e. the motor cannot be de-coupled from the driven equipment), or • if flux reduction is not allowed while the motor is running (i.e. in case of a motor with an integrated brake supplied from the motor terminals). With Reduced ID run, the control in the field weakening area or at high torques is not necessarily as accurate as with the Normal ID run. Reduced ID run is completed faster than the Normal ID Run (< 90 seconds). **Note:** Check the direction of rotation of the motor before starting the ID run. During the run, the motor will rotate in the forward direction. **WARNING!** The motor will run at up to approximately 50…100% of the nominal speed during the ID run. ENSURE THAT IT IS SAFE TO RUN THE MOTOR BEFORE PERFORMING THE ID RUN!

	(3) Standstill	Standstill ID run. The motor is injected with DC current. With asynchronous motor, the motor shaft is not rotating (with permanent magnet motor the shaft can rotate < 0.5 revolution). **Note:** This mode should be selected only if the Normal or Reduced ID run is not possible due to the restrictions caused by the connected mechanics (e.g. with lift or crane applications).
	(4) Autophasing	During autophasing, the start angle of the motor is determined. Note that other motor model values are not updated. See also parameter 11.07 AUTOPHASING MODE, and section *Autophasing* on page 38. **Notes:** • Autophasing can only be selected after the Normal/Reduced/ Standstill ID run has been performed once. Autophasing is used when an absolute encoder, a resolver or an encoder with commutation signals has been added/changed to a permanent magnet motor and there is no need to perform the Normal/Reduced/ Standstill ID run again. • During Autophasing the motor shaft must NOT be locked and the load torque must be < 5%.
	(5) Cur meas cal	Current offset and gain measurement calibration. The calibration will be performed at next start.

Parameter data

What this chapter contains

This chapter lists the parameters of the drive with some additional data. For the parameter descriptions, see chapter Parameters and firmware blocks.

Terms

Term	Definition
Actual signal	Signal measured or calculated by the drive. Can be monitored by the user. No user setting is possible.
Def	Default value
enum	Enumerated list, i.e. selection list
FbEq	Fieldbus equivalent: The scaling between the value shown on the panel and the integer used in serial communication.
Page no.	Page number for more information
INT32	32-bit integer value (31 bits + sign)
Bit pointer	Bit pointer. A bit pointer points to a single bit in the value of another parameter.
Val pointer	Value pointer. A value pointer points to the value of another parameter.
Parameter	An operation instruction of the drive that is often user-adjustable. Parameters that are signals measured or calculated by the drive are called actual signals.
Pb	Packed boolean
PT	Parameter protection type. See WP, WPD and WP0.
REAL	16-bit value 16-bit value (31 bits + sign) = integer value = fractional value
REAL24	8-bit value 24-bit value (31 bits + sign) = integer value = fractional value
Save PF	Parameter setting is protected against power failure.
Type	Data type. See enum, INT32, Bit pointer, Val pointer, Pb, REAL, REAL24, UINT32.
UINT32	32-bit unsigned integer value
WP	Write protected parameter (i.e. read only)
WPD	Write protected parameter while drive is running
WP0	Parameter can only be set to zero.

Fieldbus equivalent

Serial communication data between fieldbus adapter and drive is transferred in integer format. Thus the drive actual and reference signal values must be scaled to 16/32-bit integer values. Fieldbus equivalent defines the scaling between the signal value and the integer used in serial communication.

All the read and sent values are limited to 16/32 bits.

Example: If 32.04 MAXIMUM TORQ REF is set from external control system, an integer value of 10 corresponds to 1%.

Fieldbus addresses

For FPBA-01 PROFIBUS Adapter, FDNA-01 DeviceNet Adapter, FCAN-01 CANopen Adapter, FSCA-01 RS-485 Modbus Adapter and FENA-01 Ethernet Adapter, see the User's Manual of the fieldbus adapter module.

Pointer parameter format in fieldbus communication

Value and bit pointer parameters are transferred between the fieldbus adapter and drive as 32-bit integer values.

32-bit integer value pointers

When a value pointer parameter is connected to the value of another parameter or signal, the format is as follows:

	Bit			
	30…31	**16…29**	**8…15**	**0…7**
Name	Source type	Not in use	Group	Index
Value	1	-	1…255	1…255
Description	Value pointer is connected to parameter/signal.	-	Group of source parameter	Index of source parameter

For example, the value that should be written into parameter 33.02 SUPERV1 ACT to change its value to 1.07 DC-VOLTAGE is
0100 0000 0000 0000 0000 0001 0000 0111 = 1073742087 (32-bit integer).

When a value pointer parameter is connected to an application program, the format is as follows:

	Bit		
	30…31	**24…29**	**0…23**
Name	Source type	Not in use	Address
Value	2	-	0 … 2^{24}-1

	Bit		
	30…31	**24…29**	**0…23**
Description	Value pointer is connected to application program.	-	Relative address of application program variable

Note: Value pointer parameters which are connected to an application program cannot be set via fieldbus (i.e. read access only).

32-bit integer bit pointers

When a bit pointer parameter is connected to value 0 or 1, the format is as follows:

	Bit		
	30…31	**16…29**	**0**
Name	Source type	Not in use	Value
Value	0	-	0…1
Description	Bit pointer is connected to 0/1.	-	0 = False, 1 = True

When a bit pointer is connected to a bit value of another signal, the format is as follows:

	Bit				
	30…31	**24…29**	**16…23**	**8…15**	**0…7**
Name	Source type	Not in use	Bit sel	Group	Index
Value	1	-	0…31	2…255	1…255
Description	Bit pointer is connected to signal bit value.	-	Bit selection	Group of source parameter	Index of source parameter

When a bit pointer parameter is connected to an application program, the format is as follows:

	Bit		
	30…31	**24…29**	**0…23**
Name	Source type	Bit sel	Address
Value	2	0…31	0 … 2^{24}-1
Description	Bit pointer is connected to application program.	Bit selection	Relative address of application program variable

Note: Bit pointer parameters which are connected to an application program cannot be set via fieldbus (i.e. read access only).

Actual signals (Parameter groups 1…9)

Index	Name	Type	Range	Unit	FbEq	Update time	Data length	PT	Save PF	Page no.
01	ACTUAL VALUES									
1.01	SPEED ACT	REAL	-30000…30000	rpm	1 = 100	250 µs	32	WP		59
1.02	SPEED ACT PERC	REAL	-1000…1000	%	1 = 100	2 ms	32	WP		59
1.03	FREQUENCY	REAL	-30000…30000	Hz	1 = 100	2 ms	32	WP		59
1.04	CURRENT	REAL	0…30000	A	1 = 100	10 ms	32	WP		59
1.05	CURRENT PERC	REAL	0…1000	%	1 = 10	2 ms	16	WP		59
1.06	TORQUE	REAL	-1600…1600	%	1 = 10	2 ms	16	WP		59
1.07	DC-VOLTAGE	REAL	-	V	1 = 100	2 ms	32	WP		59
1.08	ENCODER 1 SPEED	REAL	-	rpm	1 = 100	250 µs	32	WP		59
1.09	ENCODER 1 POS	REAL24	-	rev	1=100000000	250 µs	32	WP		59
1.10	ENCODER 2 SPEED	REAL	-	rpm	1 = 100	250 µs	32	WP		60
1.11	ENCODER 2 POS	REAL24	-	rev	1=100000000	250 µs	32	WP		60
1.14	SPEED ESTIMATED	REAL	-30000…30000	rpm	1 = 100	2 ms	32	WP		60
1.15	TEMP INVERTER	REAL24	-40…160	°C	1 = 10	2 ms	16	WP		60
1.16	TEMP BC	REAL24	-40…160	°C	1 = 10	2 ms	16	WP		60
1.17	MOTOR TEMP	REAL	-10…250	°C	1 = 10	10 ms	16	WP		60
1.18	MOTOR TEMP EST	INT32	-60…1000	°C	1 = 1	-	16	WP	x	60
1.19	USED SUPPLY VOLT	REAL	0…1000	V	1 = 10	10 ms	16	WP		60
1.20	BRAKE RES LOAD	REAL24	0…1000	%	1 = 1	50 ms	16	WP		60
1.21	CPU USAGE	UINT32	0…100	%	1 = 1	-	16	WP		60
1.22	INVERTER POWER	REAL	-2^{31}…2^{31} - 1	kW	1 = 100	10 ms	32	WP		60
1.26	ON TIME COUNTER	INT32	0…35791394.1	h	1 = 100	10 ms	32	WP0	x	60
1.27	RUN TIME COUNTER	INT32	0…35791394.1	h	1 = 100	10 ms	32	WP0	x	60
1.28	FAN ON-TIME	INT32	0…35791394.1	h	1 = 100	10 ms	32	WP0	x	60
1.31	MECH TIME CONST	REAL	0…32767	s	1 = 1000	10 ms	32	WP	x	61
02	I/O VALUES									
2.01	DI STATUS	Pb	0…0x3F	-	1 = 1	2 ms	16	WP		62
2.02	RO STATUS	Pb	-	-	1 = 1	2 ms	16	WP		62
2.03	DIO STATUS	Pb	-	-	1 = 1	2 ms	16	WP		62
2.04	AI1	REAL	-	V or mA	1 = 1000	2 ms	16	WP		62
2.05	AI1 SCALED	REAL	-	-	1 = 1000	250 µs	32	WP		62
2.06	AI2	REAL	-	V or mA	1 = 1000	2 ms	16	WP		62
2.07	AI2 SCALED	REAL	-	-	1 = 1000	250 µs	32	WP		62
2.08	AO1	REAL	-	mA	1 = 1000	2 ms	16	WP		62
2.09	AO2	REAL	-	V	1 = 1000	2 ms	16	WP		62
2.10	DIO2 FREQ IN	REAL	0…32767	Hz	1 = 1000	2 ms	32	WP		62
2.11	DIO3 FREQ OUT	REAL	0…32767	Hz	1 = 1000	2 ms	32	WP		62
2.12	FBA MAIN CW	Pb	0 … 0xFFFFFFFF	-	1 = 1	500 µs	32	WP		63
2.13	FBA MAIN SW	Pb	0 … 0xFFFFFFFF	-	1 = 1	500 µs	32	WP		65
2.14	FBA MAIN REF1	INT32	-2^{31}…2^{31} - 1	-	1 = 1	500 µs	32	WP		66
2.15	FBA MAIN REF2	INT32	-2^{31}…2^{31} - 1	-	1 = 1	500 µs	32	WP		66

Index	Name	Type	Range	Unit	FbEq	Update time	Data length	PT	Save PF	Page no.
2.16	FEN DI STATUS	Pb	0…0x33	-	1 = 1	500 µs	16	WP		66
2.17	D2D MAIN CW	Pb	0…0xFFFF	-	1 = 1	500 µs	16	WP		66
2.18	D2D FOLLOWER CW	Pb	0…0xFFFF	-	1 = 1	2 ms	16	WP		67
2.19	D2D REF1	REAL	-2^{31}…2^{31} - 1	-	1 = 1	500 µs	32	WP		67
2.20	D2D REF2	REAL	-2^{31}…2^{31} - 1	-	1 = 1	2 ms	32	WP		67
03	CONTROL VALUES									
3.01	SPEED REF1	REAL	-30000…30000	rpm	1 = 100	500 µs	32	WP		68
3.02	SPEED REF2	REAL	-30000…30000	rpm	1 = 100	500 µs	32	WP		68
3.03	SPEEDREF RAMP IN	REAL	-30000…30000	rpm	1 = 100	500 µs	32	WP		68
3.04	SPEEDREF RAMPED	REAL	-30000…30000	rpm	1 = 100	500 µs	32	WP		68
3.05	SPEEDREF USED	REAL	-30000…30000	rpm	1 = 100	250 µs	32	WP		68
3.06	SPEED ERROR FILT	REAL	-30000…30000	rpm	1 = 100	250 µs	32	WP		68
3.07	ACC COMP TORQ	REAL	-1600…1600	%	1 = 10	250 µs	16	WP		68
3.08	TORQ REF SP CTRL	REAL	-1600…1600	%	1 = 10	250 µs	16	WP		68
3.09	TORQ REF1	REAL	-1000…1000	%	1 = 10	500 µs	16	WP		68
3.10	TORQ REF RAMPED	REAL	-1000…1000	%	1 = 10	500 µs	16	WP		68
3.11	TORQ REF RUSHLIM	REAL	-1000…1000	%	1 = 10	250 µs	16	WP		68
3.12	TORQUE REF ADD	REAL	-1000…1000	%	1 = 10	250 µs	16	WP		68
3.13	TORQ REF TO TC	REAL	-1600…1600	%	1 = 10	250 µs	16	WP		68
3.14	BRAKE TORQ MEM	REAL	-1000…1000	%	1 = 10	2 ms	16	WP	x	68
3.15	BRAKE COMMAND	enum	0…1	-	1 = 1	2 ms	16	WP		69
3.16	FLUX REF USED	REAL24	0…200	%	1 = 1	2 ms	16	WP		69
3.17	TORQUE REF USED	REAL	-1600…1600	%	1 = 10	250 µs	32	WP		69
06	DRIVE STATUS									
6.01	STATUS WORD 1	Pb	0…65535	-	1 = 1	2 ms	16	WP		70
6.02	STATUS WORD 2	Pb	0…65535	-	1 = 1	2 ms	16	WP		71
6.03	SPEED CTRL STAT	Pb	0…31	-	1 = 1	250 µs	16	WP		72
6.05	LIMIT WORD 1	Pb	0…255	-	1 = 1	250 µs	16	WP		72
6.07	TORQ LIM STATUS	Pb	0…65535	-	1 = 1	250 µs	16	WP		73
6.12	OP MODE ACK	enum	0…11	-	1 = 1	2 ms	16	WP		73
6.14	SUPERV STATUS	Pb	0…65535	-	1 = 1	2 ms	16	WP		73
08	ALARMS & FAULTS									
8.01	ACTIVE FAULT	enum	0…65535	-	1 = 1	-	16	WP		74
8.02	LAST FAULT	enum	0…65535	-	1 = 1	-	16	WP		74
8.03	FAULT TIME HI	INT32	-2^{31}…2^{31} - 1	days	1 = 1	-	32	WP		74
8.04	FAULT TIME LO	INT32	-2^{31}…2^{31} - 1	time	1 = 1	-	32	WP		74
8.05	ALARM WORD 1	UINT32	-	-	1 = 1	2 ms	16	WP0		74
8.06	ALARM WORD 2	UINT32	-	-	1 = 1	2 ms	16	WP0		75
8.07	ALARM WORD 3	UINT32	-	-	1 = 1	2 ms	16	WP0		75
8.08	ALARM WORD 4	UINT32	-	-	1 = 1	2 ms	16	WP0		76
8.09	ALARM WORD 5	UINT32	-	-	1 = 1	2 ms	16	WP0		76
8.10	ALARM WORD 6	UINT32	-	-	1 = 1	2 ms	16	WP0		76
09	SYSTEM INFO									
9.01	DRIVE TYPE	INT32	0…65535	-	1 = 1	-	16	WP		77

Index	Name	Type	Range	Unit	FbEq	Update time	Data length	PT	Save PF	Page no.
9.02	DRIVE RATING ID	INT32	0…65535	-	1 = 1	-	16	WP		77
9.03	FIRMWARE ID	Pb	-	-	1 = 1	-	16	WP		77
9.04	FIRMWARE VER	Pb	-	-	1 = 1	-	16	WP		77
9.05	FIRMWARE PATCH	Pb	-	-	1 = 1	-	16	WP		77
9.10	INT LOGIC VER	Pb	-	-	1 = 1	-	32	WP		77
9.20	OPTION SLOT 1	INT32	0…18	-	1 = 1	-	16	WP		77
9.21	OPTION SLOT 2	INT32	0…18	-	1 = 1	-	16	WP		77
9.22	OPTION SLOT 3	INT32	0…18	-	1 = 1	-	16	WP		77

Parameter groups 10…99

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
10	START/STOP										
10.01	EXT1 START FUNC	enum	0…6	-	-	2 ms	16	1	WPD		79
10.02	EXT1 START IN1	Bit pointer		-		2 ms	32	P.02.01.00	WPD		79
10.03	EXT1 START IN2	Bit pointer		-		2 ms	32	C.False	WPD		80
10.04	EXT2 START FUNC	enum	0…6	-	-	2 ms	16	1	WPD		80
10.05	EXT2 START IN1	Bit pointer		-		2 ms	32	P.02.01.00	WPD		81
10.06	EXT2 START IN2	Bit pointer		-		2 ms	32	C.False	WPD		81
10.07	JOG1 START	Bit pointer		-		2 ms	32	C.False	WPD		81
10.08	FAULT RESET SEL	Bit pointer		-		2 ms	32	P.02.01.02			81
10.09	RUN ENABLE	Bit pointer		-		2 ms	32	C.True	WPD		81
10.10	EM STOP OFF3	Bit pointer		-		2 ms	32	C.True	WPD		81
10.11	EM STOP OFF1	Bit pointer		-		2 ms	32	C.True	WPD		82
10.12	START INHIBIT	enum	0…1	-	1 = 1	2 ms	16	0			82
10.13	FB CW USED	Val pointer		-		2 ms	32	P.02.12	WPD		82
10.14	JOG2 START	Bit pointer		-		2 ms	32	C.False	WPD		82
10.15	JOG ENABLE	Bit pointer		-		2 ms	32	C.False	WPD		82
10.16	D2D CW USED	Val pointer		-		2 ms	32	P.02.17	WPD		82
10.17	START ENABLE	Bit pointer		-		2 ms	32	C.True	WPD		83
11	START/STOP MODE										
11.01	START MODE	enum	0…2	-	1 = 1	-	16	1	WPD		84
11.02	DC MAGN TIME	UINT32	0…10000	ms	1 = 1	-	16	500	WPD		85
11.03	STOP MODE	enum	1…2	-	1 = 1	2 ms	16	2			85
11.04	DC HOLD SPEED	REAL	0…1000	rpm	1 = 10	2 ms	16	5			85
11.05	DC HOLD CUR REF	UINT32	0…100	%	1 = 1	2 ms	16	30			85
11.06	DC HOLD	enum	0…1	-	1 = 1	2 ms	16	0			86
11.07	AUTOPHASING MODE	enum	0…2	-	1 = 1	-	16	1			86
12	DIGITAL IO										
12.01	DIO1 CONF	enum	0…1	-	1 = 1	10 ms	16	0			87
12.02	DIO2 CONF	enum	0…2	-	1 = 1	10 ms	16	0			88
12.03	DIO3 CONF	enum	0…3	-	1 = 1	10 ms	16	0			88
12.04	DIO1 OUT PTR	Bit pointer		-		10 ms	32	P.06.02.02			88

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
12.05	DIO2 OUT PTR	Bit pointer		-		10 ms	32	P.06.02.03			88
12.06	DIO3 OUT PTR	Bit pointer		-		10 ms	32	P.06.01.10			88
12.07	DIO3 F OUT PTR	Val pointer		-		10 ms	32	P.01.01			88
12.08	DIO3 F MAX	REAL	3…32768	Hz	1 = 1	10 ms	16	1000			88
12.09	DIO3 F MIN	REAL	3…32768	Hz	1 = 1	10 ms	16	3			88
12.10	DIO3 F MAX SCALE	REAL	0…32768	-	1 = 1	10 ms	16	1500			89
12.11	DIO3 F MIN SCALE	REAL	0…32768	-	1 = 1	10 ms	16	0			89
12.12	RO1 OUT PTR	Bit pointer		-		10 ms	32	P.03.15.00			89
12.13	DI INVERT MASK	UINT32	0…63	-	1 = 1	10 ms	16	0			89
12.14	DIO2 F MAX	REAL	3…32768	Hz	1 = 1	10 ms	16	1000			90
12.15	DIO2 F MIN	REAL	3…32768	Hz	1 = 1	10 ms	16	3			90
12.16	DIO2 F MAX SCALE	REAL	-32768… 32768	-	1 = 1	10 ms	16	1500			90
12.17	DIO2 F MIN SCALE	REAL	-32768… 32768	-	1 = 1	10 ms	16	0			90
13	ANALOGUE INPUTS										
13.01	AI1 FILT TIME	REAL	0…30	s	1 = 1000	10 ms	16	0			91
13.02	AI1 MAX	REAL	-11…11/ -22…22	V or mA	1 = 1000	10 ms	16	10			91
13.03	AI1 MIN	REAL	-11…11/ -22…22	V or mA	1 = 1000	10 ms	16	-10			92
13.04	AI1 MAX SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	1500			92
13.05	AI1 MIN SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	-1500			92
13.06	AI2 FILT TIME	REAL	0…30	s	1 = 1000	10 ms	16	0			92
13.07	AI2 MAX	REAL	-11…11/ -22…22	V or mA	1 = 1000	10 ms	16	10			93
13.08	AI2 MIN	REAL	-11…11/ -22…22	V or mA	1 = 1000	10 ms	16	-10			93
13.09	AI2 MAX SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	100			93
13.10	AI2 MIN SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	-100			93
13.11	AITUNE	enum	0…4	-	1 = 1	10 ms	16	0			93
13.12	AI SUPERVISION	enum	0…3	-	1 = 1	2 ms	16	0			94
13.13	AI SUPERVIS ACT	UINT32	0000… 1111	-	1 = 1	2 ms	32	0			94
15	ANALOGUE OUTPUTS										
15.01	AO1 PTR	Val pointer		-		-	32	P.01.05			95
15.02	AO1 FILT TIME	REAL	0…30	s	1 = 1000	10 ms	16	0.1			95
15.03	AO1 MAX	REAL	0…22.7	mA	1 = 1000	10 ms	16	20			95
15.04	AO1 MIN	REAL	0…22.7	mA	1 = 1000	10 ms	16	4			96
15.05	AO1 MAX SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	100			96

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
15.06	AO1 MIN SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	0			96
15.07	AO2 PTR	Val pointer		-		-	32	P.01.02			96
15.08	AO2 FILT TIME	REAL	0…30	s	1 = 1000	10 ms	16	0.1			96
15.09	AO2 MAX	REAL	-10…10	V	1 = 1000	10 ms	16	10			96
15.10	AO2 MIN	REAL	-10…10	V	1 = 1000	10 ms	16	-10			97
15.11	AO2 MAX SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	100			97
15.12	AO2 MIN SCALE	REAL	-32768… 32767	-	1 = 1000	10 ms	32	-100			97
16	SYSTEM										
16.01	LOCAL LOCK	Bit pointer		-		2 ms	32	C.False			98
16.02	PARAMETER LOCK	enum	0…2	-	1 = 1	2 ms	16	1			98
16.03	PASS CODE	INT32	0…2^{31} -1	-	1 = 1	-	32	0			98
16.04	PARAM RESTORE	enum	0…2	-	1 = 1	-	16	0	WPD		98
16.07	PARAM SAVE	enum	0…1	-	1 = 1	-	16	0			98
16.09	USER SET SEL	enum	1…10	-	1 = 1	-	32	1	WPD		99
16.10	USER SET LOG	Pb	0…0x7FF	-	1 = 1	-	32	0	WP		99
16.11	USER IO SET LO	Bit pointer		-		-	32	C.False			100
16.12	USER IO SET HI	Bit pointer		-		-	32	C.False			100
16.13	TIME SOURCE PRIO	enum	0…8	-	1 = 1	-	16	0			100
17	PANEL DISPLAY										
17.01	SIGNAL1 PARAM	INT32	00.00… 255.255	-	1 = 1		16	01.03			101
17.02	SIGNAL2 PARAM	INT32	00.00… 255.255	-	1 = 1		16	01.04			101
17.03	SIGNAL3 PARAM	INT32	00.00… 255.255	-	1 = 1		16	01.06			101
20	LIMITS										
20.01	MAXIMUM SPEED	REAL	0…30000	rpm	1 = 1	2 ms	32	1500			102
20.02	MINIMUM SPEED	REAL	-30000…0	rpm	1 = 1	2 ms	32	-1500			102
20.03	POS SPEED ENA	Bit pointer		-		2 ms	32	C.True			102
20.04	NEG SPEED ENA	Bit pointer		-		2 ms	32	C.True			103
20.05	MAXIMUM CURRENT	REAL	0…30000	A	1 = 100	10 ms	32	-			103
20.06	MAXIMUM TORQUE	REAL	0…1600	%	1 = 10	2 ms	16	300			103
20.07	MINIMUM TORQUE	REAL	-1600…0	%	1 = 10	2 ms	16	-300			103
20.08	THERM CURR LIM	enum	0…1	-	1 = 1	-	16	1			103
22	SPEED FEEDBACK										
22.01	SPEED FB SEL	enum	0…2	-	1 = 1	10 ms	16	0			105
22.02	SPEED ACT FTIME	REAL	0…10000	ms	1 = 1000	10 ms	32	3			105
22.03	MOTOR GEAR MUL	INT32	-2^{31}…2^{31} -1	-	1 = 1	10 ms	32	1			106
22.04	MOTOR GEAR DIV	UINT32	1…2^{31} -1	-	1 = 1	10 ms	32	1			106
22.05	ZERO SPEED LIMIT	REAL	0…30000	rpm	1 = 1000	2 ms	32	30			106
22.06	ZERO SPEED DELAY	UINT32	0…30000	ms	1 = 1	2 ms	16	0			107
22.07	ABOVE SPEED LIM	REAL	0…30000	rpm	1 = 1	2 ms	16	0			107

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
22.08	SPEED TRIPMARGIN	REAL	0…10000	rpm	1 = 10	2 ms	32	500			108
22.09	SPEED FB FAULT	enum	0…2	-	1 = 1	10 ms	16	0			108
24	SPEED REF MOD										
24.01	SPEED REF1 SEL	enum	0…8	-	1 = 1	10 ms	16	1			110
24.02	SPEED REF2 SEL	enum	0…8	-	1 = 1	10 ms	16	0			111
24.03	SPEED REF1 IN	Val pointer		-		10 ms	32	P.03.01			111
24.04	SPEED REF2 IN	Val pointer		-		10 ms	32	P.03.02			111
24.05	SPEED REF 1/2SEL	Bit pointer		-		2 ms	32	C.False			111
24.06	SPEED SHARE	REAL	-8…8	-	1 = 1000	2 ms	16	1			111
24.07	SPEEDREF NEG ENA	Bit pointer		-		2 ms	32	C.False			112
24.08	CONST SPEED	REAL	-30000…. 30000	rpm	1 = 1	2 ms	16	0			112
24.09	CONST SPEED ENA	Bit pointer		-		2 ms	32	C.False			112
24.10	SPEED REF JOG1	REAL	-30000…. 30000	rpm	1 = 1	2 ms	16	0			112
24.11	SPEED REF JOG2	REAL	-30000…. 30000	rpm	1 = 1	2 ms	16	0			112
24.12	SPEED REFMIN ABS	REAL	0…30000	rpm	1 = 1	2 ms	16	0			112
25	SPEED REF RAMP										
25.01	SPEED RAMP IN	Val pointer		-		10 ms	32	P.03.03	WP		114
25.02	SPEED SCALING	REAL	0…30000	rpm	1 = 1	10 ms	16	1500			114
25.03	ACC TIME	REAL	0…1800	s	1 = 1000	10 ms	32	1			114
25.04	DEC TIME	REAL	0…1800	s	1 = 1000	10 ms	32	1			115
25.05	SHAPE TIME ACC1	REAL	0…1000	s	1 = 1000	10 ms	32	0			115
25.06	SHAPE TIME ACC2	REAL	0…1000	s	1 = 1000	10 ms	32	0			115
25.07	SHAPE TIME DEC1	REAL	0…1000	s	1 = 1000	10 ms	32	0			115
25.08	SHAPE TIME DEC2	REAL	0…1000	s	1 = 1000	10 ms	32	0			116
25.09	ACC TIME JOGGING	REAL	0…1800	s	1 = 1000	10 ms	32	0			116
25.10	DEC TIME JOGGING	REAL	0…1800	s	1 = 1000	10 ms	32	0			116
25.11	EM STOP TIME	REAL	0…1800	s	1 = 1000	10 ms	32	1			116
25.12	SPEEDREF BAL	REAL	-30000… 30000	rpm	1 = 1000	2 ms	32	0			116
25.13	SPEEDREF BAL ENA	Bit pointer		-		2 ms	32	C.False			116
26	SPEED ERROR										
26.01	SPEED ACT NCTRL	Val pointer		-		2 ms	32	P.01.01	WP		118
26.02	SPEED REF NCTRL	Val pointer		-		2 ms	32	P.03.04	WP		118
26.03	SPEED REF PCTRL	Val pointer		-		2 ms	32	P.04.01			118
26.04	SPEED FEED PCTRL	Val pointer		-		2 ms	32	P.04.20			118
26.05	SPEED STEP	REAL	-30000… 30000	rpm	1 = 100	2 ms	32	0			119
26.06	SPD ERR FTIME	REAL	0…1000	ms	1 = 10	2 ms	16	0			119
26.07	SPEED WINDOW	REAL	0…30000	rpm	1 = 1	250 µs	16	100			119
26.08	ACC COMP DERTIME	REAL	0…600	s	1 = 100	2 ms	32	0			119
26.09	ACC COMP FTIME	REAL	0…1000	ms	1 = 10	2 ms	16	8			120

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
26.10	SPEED WIN FUNC	UINT32	0…2	-	1 = 1	250 µs	16	0			120
26.11	SPEED WIN HI	REAL	0…3000	rpm	1 = 1	250 µs	16	0		x	120
26.12	SPEED WIN LO	REAL	0…3000	rpm	1 = 1	250 µs	16	0		x	120
28	SPEED CONTROL										
28.01	SPEED ERR NCTRL	Val pointer		-		2 ms	32	P.03.06	WP		122
28.02	PROPORT GAIN	REAL	0…200	-	1 = 100	2 ms	16	10			122
28.03	INTEGRATION TIME	REAL	0…600	s	1 = 1000	2 ms	32	0.5			123
28.04	DERIVATION TIME	REAL	0…10	s	1 = 1000	2 ms	16	0			124
28.05	DERIV FILT TIME	REAL	0…1000	ms	1 = 10	2 ms	16	8			124
28.06	ACC COMPENSATION	Val pointer		-		2 ms	32	P.03.07	WP		124
28.07	DROOPING RATE	REAL	0…100	%	1 = 100	2 ms	16	0			125
28.08	BAL REFERENCE	REAL	-1600… 1600	%	1 = 10	2 ms	16	0			125
28.09	SPEEDCTRL BAL EN	Bit pointer		-		2 ms	32	C.False			125
28.10	MIN TORQ SP CTRL	REAL	-1600… 1600	%	1 = 10	2 ms	16	-300			125
28.11	MAX TORQ SP CTRL	REAL	-1600… 1600	%	1 = 10	2 ms	16	300			125
28.12	PI ADAPT MAX SPD	REAL	0…30000	rpm	1 = 1	10 ms	16	0			126
28.13	PI ADAPT MIN SPD	REAL	0…30000	rpm	1 = 1	10 ms	16	0			126
28.14	P GAIN ADPT COEF	REAL	0…10	-	1 = 1000	10 ms	16	0			126
28.15	I TIME ADPT COEF	REAL	0…10	-	1 = 1000	10 ms	16	0			126
28.16	PI TUNE MODE	enum	0…4	-	1 = 1		16	0			127
28.17	TUNE BANDWIDTH	REAL	0…2000	Hz	1 = 100		16	100			127
28.18	TUNE DAMPING	REAL	0…200	-	1 = 10		16	0.5			127
32	TORQUE REFERENCE										
32.01	TORQ REF1 SEL	enum	0…4	-	1 = 1	10 ms	16	2			129
32.02	TORQ REF ADD SEL	enum	0…4	-	1 = 1	10 ms	16	0			129
32.03	TORQ REF IN	Val pointer		-		250 µs	32	P.03.09			130
32.04	MAXIMUM TORQ REF	REAL	0…1000	%	1 = 10	250 µs	16	300			130
32.05	MINIMUM TORQ REF	REAL	-1000…0	%	1 = 10	250 µs	16	-300			130
32.06	LOAD SHARE	REAL	-8…8	-	1 = 1000	250 µs	16	1			130
32.07	TORQ RAMP UP	UINT32	0…60	s	1 = 1000	10 ms	32	0			130
32.08	TORQ RAMP DOWN	UINT32	0…60	s	1 = 1000	10 ms	32	0			130
33	SUPERVISION										
33.01	SUPERV1 FUNC	UINT32	0…4	-	1 = 1	2 ms	16	0			132
33.02	SUPERV1 ACT	Val pointer		-		2 ms	32	P.01.01			132
33.03	SUPERV1 LIM HI	REAL	-32768… 32768	-	1 = 100	2 ms	32	0			133
33.04	SUPERV1 LIM LO	REAL	-32768… 32768	-	1 = 100	2 ms	32	0			133
33.05	SUPERV2 FUNC	UINT32	0…4	-	1 = 1	2 ms	16	0			133
33.06	SUPERV2 ACT	Val pointer		-		2 ms	32	P.01.04			133

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
33.07	SUPERV2 LIM HI	REAL	-32768… 32768	-	1 = 100	2 ms	32	0			133
33.08	SUPERV2 LIM LO	REAL	-32768… 32768	-	1 = 100	2 ms	32	0			133
33.09	SUPERV3 FUNC	UINT32	0…4	-	1 = 1	2 ms	16	0			134
33.10	SUPERV3 ACT	Val pointer		-		2 ms	32	P.01.06			134
33.11	SUPERV3 LIM HI	REAL	-32768… 32768	-	1 = 100	2 ms	32	0			134
33.12	SUPERV3 LIM LO	REAL	-32768… 32768	-	1 = 100	2 ms	32	0			134
34	REFERENCE CTRL										
34.01	EXT1/EXT2 SEL	Bit pointer		-		2 ms	32	P.02.01.01			136
34.02	EXT1 MODE 1/2SEL	Bit pointer		-		2 ms	32	C.False (P.02.01.05 for pos. appl.)			136
34.03	EXT1 CTRL MODE1	enum	1…5 (1…9 for pos. appl.)	-	1 = 1	2 ms	16	1			136
34.04	EXT1 CTRL MODE2	enum	1…5 (1…9 for pos. appl.)	-	1 = 1	2 ms	16	2 (8 for pos. appl.)			137
34.05	EXT2 CTRL MODE1	enum	1…5 (1…9 for pos. appl.)	-	1 = 1	2 ms	16	2 (6 for pos. appl.)			137
34.07	LOCAL CTRL MODE	enum	1…2 (1…6 for pos. appl.)	-	1 = 1	2 ms	16	1	WPD		137
34.08	TREF SPEED SRC	Val pointer		-		250 µs	32	P.03.08	WP		137
34.09	TREF TORQ SRC	Val pointer		-		250 µs	32	P.03.11	WP		137
34.10	TORQ REF ADD SRC	Val pointer		-		250 µs	32	P.03.12	WP		138
35	MECH BRAKE CTRL										
35.01	BRAKE CONTROL	enum	0…2	-	1 = 1	2 ms	16	0	WPD		139
35.02	BRAKE ACKNOWL	Bit pointer		-		2 ms	32	C.False	WPD		139
35.03	BRAKE OPEN DELAY	UINT32	0…5	s	1 = 100	2 ms	16	0			140
35.04	BRAKE CLOSE DLY	UINT32	0…60	s	1 = 100	2 ms	16	0			140
35.05	BRAKE CLOSE SPD	REAL	0…1000	rpm	1 = 10	2 ms	16	100			140
35.06	BRAKE OPEN TORQ	REAL	0…1000	%	1 = 10	2 ms	16	0			140
35.07	BRAKE CLOSE REQ	Bit pointer		-		2 ms	32	C.False	WPD		140
35.08	BRAKE OPEN HOLD	Bit pointer		-		2 ms	32	C.False	WPD		140
35.09	BRAKE FAULT FUNC	enum	0…2	-	1 = 1	2 ms	16	0			140
40	MOTOR CONTROL										
40.01	FLUX REF	REAL	0…200	%	1 = 1	10 ms	16	100			142
40.02	SF REF	enum	0…16	kHz	1 = 1	-	16	4			142
40.03	SLIP GAIN	REAL	0…200	%	1 = 1	-		100			143
40.04	VOLTAGE RESERVE	REAL		V/%	1 = 1	-		-			143
40.05	FLUX OPT	enum	0…1	-	1 = 1	-		-			143

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
40.06	FORCE OPEN LOOP	enum	0…1	-	1 = 1	250 µs	16	0			143
40.07	IR COMPENSATION	REAL24	0…50	%	1 = 100	2 ms	32	0			143
45	MOT THERM PROT										
45.01	MOT TEMP PROT	enum	0…2	-	1 = 1	10 ms	16	2			145
45.02	MOT TEMP SOURCE	enum	0…6	-	1 = 1	10 ms	16	0			145
45.03	MOT TEMP ALM LIM	INT32	0…200	°C	1 = 1	-	16	90			146
45.04	MOT TEMP FLT LIM	INT32	0…200	°C	1 = 1	-	16	110			146
45.05	AMBIENT TEMP	INT32	-60…100	°C	1 = 1	-	16	20			146
45.06	MOT LOAD CURVE	INT32	50…150	%	1 = 1	-	16	100			147
45.07	ZERO SPEED LOAD	INT32	50…150	%	1 = 1	-	16	100			147
45.08	BREAK POINT	INT32	0.01…500	Hz	1 = 100	-	16	45			147
45.09	MOTNOM TEMP RISE	INT32	0…300	°C	1 = 1	-	16	80			148
45.10	MOT THERM TIME	INT32	100…10000	s	1 = 1	-	16	256			148
46	FAULT FUNCTIONS										
46.01	EXTERNAL FAULT	Bit pointer		-		2 ms	32	C.True			149
46.02	SPEED REF SAFE	REAL	-30000… 30000	rpm	1 = 1	2 ms	16	0			149
46.03	LOCAL CTRL LOSS	enum	0…3	-	1 = 1	-	16	1			149
46.04	MOT PHASE LOSS	enum	0…1	-	1 = 1	2 ms	16	1			150
46.05	EARTH FAULT	enum	0…2	-	1 = 1	-	16	2			150
46.06	SUPPL PHS LOSS	enum	0…1	-	1 = 1	2 ms	16	1			150
46.07	STO DIAGNOSTIC	enum	1…4	-	1 = 1	10 ms	16	1			150
46.08	CROSS CONNECTION	enum	0…1	-	1 = 1	-	16	1			151
47	VOLTAGE CTRL										
47.01	OVERVOLTAGE CTRL	enum	0…1	-	1 = 1	10 ms	16	1			152
47.02	UNDERVOLT CTRL	enum	0…1	-	1 = 1	10 ms	16	1			152
47.03	SUPPLVOLTAUTO-ID	enum	0…1	-	1 = 1	10 ms	16	1			152
47.04	SUPPLY VOLTAGE	REAL	0…1000	V	1 = 10	2 ms	16	400			153
47.05	LOW VOLT MOD ENA	Bit pointer		-			32	C.False			153
47.06	LOW VOLT DC MIN	REAL	250…450	V	1 = 1	10 ms	16	250			153
47.07	LOW VOLT DC MAX	REAL	350…810	V	1 = 1	10 ms	16	810			153
47.08	EXT PU SUPPLY	Bit pointer		-			32	C.False			153
48	BRAKE CHOPPER										
48.01	BC ENABLE	enum	0…2	-	1 = 1	-	16	0			154
48.02	BC RUN-TIME ENA	Bit pointer		-		2 ms	32	C.True			154
48.03	BRTHERMTIMECONST	REAL24	0…10000	s	1 = 1	-	32	0			154
48.04	BR POWER MAX CNT	REAL24	0…10000	kW	1 = 10000	-	32	0			154
48.05	R BR	REAL24	0.1…1000	ohm	1 = 10000	-	32	-			154
48.06	BR TEMP FAULTLIM	REAL24	0…150	%	1 = 1	-	16	105			155
48.07	BR TEMP ALARMLIM	REAL24	0…150	%	1 = 1	-	16	95			155

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
50	FIELDBUS										
50.01	FBA ENABLE	enum	0…1	-	1 = 1	-	16	0			156
50.02	COMM LOSS FUNC	enum	0…3	-	1 = 1	-	16	0			156
50.03	COMM LOSS T OUT	UINT32	0.3…6553.5	s	1 = 10	-	16	0.3			157
50.04	FBA REF1 MODESEL	enum	0…2 (0…4 for pos. appl.)	-	1 = 1	10 ms	16	2			157
50.05	FBA REF2 MODESEL	enum	0…2 (0…4 for pos. appl.)	-	1 = 1	10 ms	16	3			157
50.06	FBA ACT1 TR SRC	Val pointer		-		10 ms	32	P.01.01			157
50.07	FBA ACT2 TR SRC	Val pointer		-		10 ms	32	P.01.06			157
50.08	FBA SW B12 SRC	Bit pointer		-		500 µs	32	C.False			158
50.09	FBA SW B13 SRC	Bit pointer		-		500 µs	32	C.False			158
50.10	FBA SW B14 SRC	Bit pointer		-		500 µs	32	C.False			158
50.11	FBA SW B15 SRC	Bit pointer		-		500 µs	32	C.False			158
51	FBA SETTINGS										
51.01	FBA TYPE	UINT32	0…65536	-	1 = 1		16	0			159
51.02	FBA PAR2	UINT32	0…65536	-	1 = 1		16	0		x	159
…	…	…	…	…	…		….	…			
51.26	FBA PAR26	UINT32	0…65536	-	1 = 1		16	0		x	159
51.27	FBA PAR REFRESH	UINT32	0…1	-	1 = 1		16	0	WPD	x	159
51.28	PAR TABLE VER	UINT32	0…65536	-	1 = 1		16	0		x	159
51.29	DRIVE TYPE CODE	UINT32	0…65536	-	1 = 1		16	0		x	159
51.30	MAPPING FILE VER	UINT32	0…65536	-	1 = 1		16	0		x	160
51.31	D2FBA COMM STA	UINT32	0…6	-	1 = 1		16	0		x	160
51.32	FBA COMM SW VER	UINT32	0…65536	-	1 = 1		16	0		x	160
51.33	FBA APPL SW VER	UINT32	0…65536	-	1 = 1		16	0		x	160
52	FBA DATA IN										
52.01	FBA DATA IN1	UINT32	0…9999	-	1 = 1		16	0		x	161
…	…	…	…	…	…		…	…			-
52.12	FBA DATA IN12	UINT32	0…9999	-	1 = 1		16	0		x	161
53	FBA DATA OUT										
53.01	FBA DATA OUT1	UINT32	0…9999	-	1 = 1		16	0		x	162
…	…	…	…	…	…		…	…			
53.12	FBA DATA OUT12	UINT32	0…9999	-	1 = 1		16	0		x	162

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
55	COMMUNICATION TOOL										
55.01	MDB STATION ID	UINT32	1…247	-	1 = 1		16	1			163
55.02	MDB BAUD RATE	UINT32	0…4	-	1 = 1		16	0			163
55.03	MDB PARITY	UINT32	0…3	-	1 = 1		16	0			163
57	D2D COMMUNICATION										
57.01	LINK MODE	UINT32	0…2	-	1 = 1	10 ms	16	0	WPD		164
57.02	COMM LOSS FUNC	UINT32	0…2	-	1 = 1	10 ms	16	1			164
57.03	NODE ADDRESS	UINT32	1…62	-	1 = 1	10 ms	16	1	WPD		164
57.04	FOLLOWER MASK 1	UINT32	0…2^{31}	-	1 = 1	10 ms	32	0	WPD		165
57.05	FOLLOWER MASK 2	UINT32	0…2^{31}	-	1 = 1	10 ms	32	0	WPD		165
57.06	REF 1 SRC	Val pointer		-		10 ms	32	P.03.04			165
57.07	REF 2 SRC	Val pointer		-		10 ms	32	P.03.13			165
57.08	FOLLOWER CW SRC	Val pointer		-		10 ms	32	P.02.18			165
57.09	KERNEL SYNC MODE	enum	0…3	-	1 = 1	10 ms	16	0	WPD		165
57.10	KERNEL SYNC OFFS	REAL	-4999… 5000	ms	1 = 1	10 ms	16	0	WPD		166
57.11	REF 1 MSG TYPE	UINT32	0…1	-	1 = 1	10 ms	16	0			166
57.12	REF1 MC GROUP	UINT32	0…62	-	1 = 1	10 ms	16	0			166
57.13	NEXT REF1 MC GRP	UINT32	0…62	-	1 = 1	10 ms	16	0			166
57.14	NR REF1 MC GRPS	UINT32	1…62	-	1 = 1	10 ms	16	1			167
57.15	D2D COMM PORT	UINT32	0…3	-	1 = 1		16	0	WPD		167
90	ENC MODULE SEL										
90.01	ENCODER 1 SEL	enum	0…6	-	1 = 1		16	0			169
90.02	ENCODER 2 SEL	enum	0…6	-	1 = 1		16	0			170
90.03	EMUL MODE SEL	enum	0…9	-	1 = 1		16	0			170
90.04	TTL ECHO SEL	enum	0…4	-	1 = 1		16	0			171
90.05	ENC CABLE FAULT	UINT32	0…2	-	1 = 1		16	1			171
90.10	ENC PAR REFRESH	UINT32	0…1	-	1 = 1		16	0	WPD		172
91	ABSOL ENC CONF										
91.01	SINE COSINE NR	UINT32	0…65535	-	1 = 1		16	0			174
91.02	ABS ENC INTERF	UINT32	0…4	-	1 = 1		16	0			174
91.03	REV COUNT BITS	UINT32	0…32	-	1 = 1		16	0			174
91.04	POS DATA BITS	UINT32	0…32	-	1 = 1		16	0			174
91.05	REFMARK ENA	UINT32	0…1	-	1 = 1		16	0			174
91.10	HIPERFACE PARITY	UINT32	0…1	-	1 = 1		16	0			174
91.11	HIPERF BAUDRATE	UINT32	0…3	-	1 = 1		16	1			175
91.12	HIPERF NODE ADDR	UINT32	0…255	-	1 = 1		16	64			175
91.20	SSI CLOCK CYCLES	UINT32	2…127	-	1 = 1		16	2			175
91.21	SSI POSITION MSB	UINT32	1…126	-	1 = 1		16	1			175
91.22	SSI REVOL MSB	UINT32	1…126	-	1 = 1		16	1			175
91.23	SSI DATA FORMAT	UINT32	0…1	-	1 = 1		16	0			175
91.24	SSI BAUD RATE	UINT32	0…5	-	1 = 1		16	2			176

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
91.25	SSI MODE	UINT32	0…1	-	1 = 1		16	0			176
91.26	SSI TRANSMIT CYC	UINT32	0…5	-	1 = 1		16	1			176
91.27	SSI ZERO PHASE	UINT32	0…3	-	1 = 1		16	0			176
91.30	ENDAT MODE	UINT32	0…1	-	1 = 1		16	0			177
91.31	ENDAT MAX CALC	UINT32	0…3	-	1 = 1		16	3			177
92	RESOLVER CONF										
92.01	RESOLV POLEPAIRS	UINT32	1…32	-	1 = 1		16	1			178
92.02	EXC SIGNAL AMPL	UINT32	4…12	Vrms	1 = 10		16	4			178
92.03	EXC SIGNAL FREQ	UINT32	1…20	kHz	1 = 1		16	1			178
93	PULSE ENC CONF										
93.01	ENC1 PULSE NR	UINT32	0…65535	-	1 = 1		16	0			179
93.02	ENC1 TYPE	enum	0…1	-	1 = 1		16	0			179
93.03	ENC1 SP CALCMODE	enum	0….5	-	1 = 1		16	4			179
93.04	ENC1 POS EST ENA	enum	0…1	-	1 = 1		16	1			180
93.05	ENC1 SP EST ENA	enum	0…1	-	1 = 1		16	0			180
93.06	ENC1 OSC LIM	enum	0…3	-	1 = 1		16	0			180
93.11	ENC2 PULSE NR	UINT32	0…65535	-	1 = 1		16	0			181
93.12	ENC2 TYPE	enum	0…1	-	1 = 1		16	0			181
93.13	ENC2 SP CALCMODE	enum	0….5	-	1 = 1		16	4			181
93.14	ENC2 POS EST ENA	enum	0…1	-	1 = 1		16	1			181
93.15	ENC2 SP EST ENA	enum	0…1	-	1 = 1		16	0			181
93.16	ENC2 OSC LIM	enum	0…3	-	1 = 1		16	0			181
93.21	EMUL PULSE NR	UINT32	0…65535	-	1 = 1		16	0			181
93.22	EMUL POS REF	Val pointer		-			32	P.01.12 (P.04.17 for pos. appl.)			181
93.23	EMUL POS OFFSET	REAL	0 … 0.99998	rev	1 = 100000		32	0			181
95	HW CONFIGURATION										
95.01	CTRL UNIT SUPPLY	enum	0…1	-	1 = 1		16	0			182
95.02	EXTERNAL CHOKE	enum	0…1	-	1 = 1		16	0			182
97	USER MOTOR PAR										
97.01	USE GIVEN PARAMS	enum	0…3	-	1 = 1		16	0	WPD		183
97.02	RS USER	REAL24	0…0.5	p.u.	1 = 100000		32	0			183
97.03	RR USER	REAL24	0…0.5	p.u.	1 = 100000		32	0			183
97.04	LM USER	REAL24	0…10	p.u.	1 = 100000		32	0			183
97.05	SIGMAL USER	REAL24	0…1	p.u.	1 = 100000		32	0			183
97.06	LD USER	REAL24	0…10	p.u.	1 = 100000		32	0			183
97.07	LQ USER	REAL24	0…10	p.u.	1 = 100000		32	0			184
97.08	PM FLUX USER	REAL24	0…2	p.u.	1 = 100000		32	0			184
97.09	RS USER SI	REAL24	0…100	ohm	1 = 100000		32	0			184
97.10	RR USER SI	REAL24	0…100	ohm	1 = 100000		32	0			184

Index	Parameter	Type	Range	Unit	FbEq	Update time	Data len.	Def	PT	Save PF	Page no.
97.11	LM USER SI	REAL24	0…100000	mH	1 = 100000		32	0			184
97.12	SIGL USER SI	REAL24	0…100000	mH	1 = 100000		32	0			184
97.13	LD USER SI	REAL24	0…100000	mH	1 = 100000		32	0			184
97.14	LQ USER SI	REAL24	0…100000	mH	1 = 100000		32	0			184
97.20	POS OFFSET USER	REAL	0…360	° (el.)	1 = 1		32	0			185
98	MOTOR CALC VALUES										
98.01	TORQ NOM SCALE	UINT32	0…2147483	Nm	1 = 1000		32	0	WP		186
98.02	POLEPAIRS	UINT32	0…1000	-	1 = 1		16	0	WP		186
99	START-UP DATA										
99.01	LANGUAGE	enum		-	1 = 1		16				187
99.04	MOTOR TYPE	enum	0…1	-	1 = 1		16	0	WPD		187
99.05	MOTOR CTRL MODE	enum	0…1	-	1 = 1		16	0			188
99.06	MOT NOM CURRENT	REAL	0…6400	A	1 = 10		32	0	WPD		188
99.07	MOT NOM VOLTAGE	REAL	120…960	V	1 = 10		32	0	WPD		188
99.08	MOT NOM FREQ	REAL	0…500	Hz	1 = 10		32	0	WPD		189
99.09	MOT NOM SPEED	REAL	0…30000	rpm	1 = 1		32	0	WPD		189
99.10	MOT NOM POWER	REAL	0…10000	kW	1 = 100		32	0	WPD		189
99.11	MOT NOM COSFII	REAL24	0…1	-	1 = 100		32	0	WPD		189
99.12	MOT NOM TORQUE	INT32	0…2147483	Nm	1 = 1000		32	0	WPD		189
99.13	IDRUN MODE	enum	0…5	-	1 = 1		16	0	WPD		190

Fault tracing

What this chapter contains

The chapter lists all alarm and fault messages including the possible cause and corrective actions.

Safety

WARNING! Only qualified electricians are allowed to maintain the drive. The *Safety Instructions* on the first pages of the appropriate hardware manual must be read before you start working with the drive.

Alarm and fault indications

An alarm or a fault message indicates abnormal drive status. Most alarm and fault causes can be identified and corrected using this information. If not, an ABB representative should be contacted.

The four-digit code number in brackets after the message is for the fieldbus communication.

The alarm/fault code is displayed on the 7-segment display of the drive. The following table describes the indications given by the 7-segment display.

Display	Meaning
“E-” followed by error code	System error. See appropriate drive hardware manual.
“A-” followed by error code	Alarm. See section *Alarm messages generated by the drive* on page 211.
“F-” followed by error code	Fault. See section *Fault messages generated by the drive* on page 219.

How to reset

The drive can be reset either by pressing the reset key on the PC tool ( ) or control panel (***RESET***) or switching the supply voltage off for a while. When the fault has been removed, the motor can be restarted.

A fault can also be reset from an external source by parameter 10.08 FAULT RESET SEL.

Fault history

When a fault is detected, it is stored in the fault logger with a time stamp. The fault history stores information on the 16 latest faults of the drive. Three of the latest faults are stored at the beginning of a power switch off.

Signals 8.01 ACTIVE FAULT and 8.02 LAST FAULT store the fault codes of the most recent faults.

Alarms can be monitored via alarm words 8.05 ALARM WORD 1 … 8.08 ALARM WORD 4. Alarm information is lost at power switch off or fault reset.

Alarm messages generated by the drive

Code	Alarm (fieldbus code)	Cause	What to do
2000	BRAKE START TORQUE (0x7185) Programmable fault: 35.09 BRAKE FAULT FUNC	Mechanical brake alarm. Alarm is activated if required motor starting torque, 35.06 BRAKE OPEN TORQ, is not achieved.	Check brake open torque setting, parameter 35.06. Check drive torque and current limits. See firmware block LIMITS on page 102.
2001	BRAKE NOT CLOSED (0x7186) Programmable fault: 35.09 BRAKE FAULT FUNC	Mechanical brake control alarm. Alarm is activated e.g. if brake acknowledgement is not as expected during brake closing.	Check mechanical brake connection. Check mechanical brake settings, parameters 35.01…35.09. To determine whether problem is with acknowledgement signal or brake: Check if brake is closed or open.
2002	BRAKE NOT OPEN (0x7187) Programmable fault: 35.09 BRAKE FAULT FUNC	Mechanical brake control alarm. Alarm is activated e.g. if brake acknowledgement is not as expected during brake opening.	Check mechanical brake connection. Check mechanical brake settings, parameters 35.01…35.08. To determine whether problem is with acknowledgement signal or brake: Check if brake is closed or open.
2003	SAFE TORQUE OFF (0xFF7A) Programmable fault: 46.07 STO DIAGNOSTIC	Safe Torque Off function is active, i.e. safety circuit signal(s) connected to connector X6 is lost while drive is stopped and parameter 46.07 STO DIAGNOSTIC is set to (2) Alarm.	Check safety circuit connections. For more information, see appropriate drive hardware manual.
2005	MOTOR TEMPERATURE (0x4310) Programmable fault: 45.01 MOT TEMP PROT	Estimated motor temperature (based on motor thermal model) has exceeded alarm limit defined by parameter 45.03 MOT TEMP ALM LIM.	Check motor ratings and load. Let motor cool down. Ensure proper motor cooling: Check cooling fan, clean cooling surfaces, etc. Check value of alarm limit. Check motor thermal model settings, parameters 45.06…45.08 and 45.10 MOT THERM TIME.
		Measured motor temperature has exceeded alarm limit defined by parameter 45.03 MOT TEMP ALM LIM.	Check that actual number of sensors corresponds to value set by parameter 45.02 MOT TEMP SOURCE. Check motor ratings and load. Let motor cool down. Ensure proper motor cooling: Check cooling fan, clean cooling surfaces, etc. Check value of alarm limit.
2006	EMERGENCY OFF (0xF083)	Drive has received emergency OFF2 command.	To restart drive, activate RUN ENABLE signal (source selected by parameter 10.09 RUN ENABLE) and start drive.

Code	Alarm (fieldbus code)	Cause	What to do
2007	RUN ENABLE (0xFF54)	No Run enable signal is received.	Check setting of parameter 10.09 RUN ENABLE. Switch signal on (e.g. in the fieldbus Control Word) or check wiring of selected source.
2008	ID-RUN (0xFF84)	Motor identification run is on.	This alarm belongs to normal start-up procedure. Wait until drive indicates that motor identification is completed.
		Motor identification is required.	This alarm belongs to normal start-up procedure. Select how motor identification should be performed, parameter 99.13 IDRUN MODE. Start identification routines by pressing Start key.
2009	EMERGENCY STOP (0xF081)	Drive has received emergency stop command (OFF1/OFF3).	Check that it is safe to continue operation. Return emergency stop push button to normal position (or adjust the fieldbus Control Word accordingly). Restart drive.
2011	BR OVERHEAT (0x7112)	Brake resistor temperature has exceeded alarm limit defined by parameter 48.07 BR TEMP ALARMLIM.	Stop drive. Let resistor cool down. Check resistor overload protection function settings, parameters 48.01…48.05. Check alarm limit setting, parameter 48.07. Check that braking cycle meets allowed limits.
2012	BC OVERHEAT (0x7181)	Brake chopper IGBT temperature has exceeded internal alarm limit.	Let chopper cool down. Check for excessive ambient temperature. Check for cooling fan failure. Check for obstructions in the air flow. Check the dimensioning and cooling of the cabinet. Check resistor overload protection function settings, parameters 48.01…48.05. Check that braking cycle meets allowed limits. Check that drive supply AC voltage is not excessive.
2013	DEVICE OVERTEMP (0x4210)	Measured drive temperature has exceeded internal alarm limit.	Check ambient conditions. Check air flow and fan operation. Check heatsink fins for dust pick-up. Check motor power against unit power.

Code	Alarm (fieldbus code)	Cause	What to do
2014	INTBOARD OVERTEMP (0x7182)	Interface board (between power unit and control unit) temperature has exceeded internal alarm limit.	Let drive cool down. Check for excessive ambient temperature. Check for cooling fan failure. Check for obstructions in the air flow. Check the dimensioning and cooling of the cabinet.
2015	BC MOD OVERTEMP (0x7183)	Input bridge or brake chopper temperature has exceeded internal alarm limit.	Let drive cool down. Check for excessive ambient temperature. Check for cooling fan failure. Check for obstructions in the air flow. Check the dimensioning and cooling of the cabinet.
2016	IGBT OVERTEMP (0x7184)	Drive temperature based on thermal model has exceeded internal alarm limit.	Check ambient conditions. Check air flow and fan operation. Check heatsink fins for dust pick-up. Check motor power against unit power.
2017	FIELDBUS COMM (0x7510) Programmable fault: 50.02 COMM LOSS FUNC	Cyclical communication between drive and fieldbus adapter module or between PLC and fieldbus adapter module is lost.	Check status of fieldbus communication. See appropriate User's Manual of fieldbus adapter module. Check fieldbus parameter settings. See parameter group 50 FIELDBUS on page 156. Check cable connections. Check if communication master can communicate.
2018	LOCAL CTRL LOSS (0x5300) Programmable fault: 46.03 LOCAL CTRL LOSS	Control panel or PC tool selected as active control location for drive has ceased communicating.	Check PC tool or control panel connection. Check control panel connector. Replace control panel in mounting platform.
2019	AI SUPERVISION (0x8110) Programmable fault: 13.12 AI SUPERVISION	Analogue input AI1 or AI2 signal has reached limit defined by parameter 13.13 AI SUPERVIS ACT.	Check analogue input AI1/2 source and connections. Check analogue input AI1/2 minimum and maximum limit settings, parameters 13.02 and 13.03 / 13.07 and 13.08.
2020	FB PAR CONF (0x6320)	The drive does not have a functionality requested by PLC, or requested functionality has not been activated.	Check PLC programming. Check fieldbus parameter settings. See parameter group 50 FIELDBUS on page 156.
2021	NO MOTOR DATA (0x6381)	Parameters in group 99 have not been set.	Check that all the required parameters in group 99 have been set. **Note:** It is normal for this alarm to appear during the start-up until the motor data is entered.

Code	Alarm (fieldbus code)	Cause	What to do
2022	ENCODER 1 FAILURE (0x7301)	Encoder 1 has been activated by parameter but the encoder interface (FEN-xx) cannot be found.	Check parameter 90.01 ENCODER 1 SEL setting corresponds to encoder interface 1 (FEN-xx) installed in drive Slot 1/2 (signal 9.20 OPTION SLOT 1 / 9.21 OPTION SLOT 2). **Note:** The new setting will only take effect after parameter 90.10 ENC PAR REFRESH is used or after the JCU control unit is powered up the next time.
2023	ENCODER 2 FAILURE (0x7381)	Encoder 2 has been activated by parameter but the encoder interface (FEN-xx) cannot be found.	Check parameter 90.02 ENCODER 2 SEL setting corresponds to encoder interface 2 (FEN-xx) installed in drive Slot 1/2 (signal 9.20 OPTION SLOT 1 / 9.21 OPTION SLOT 2). **Note:** The new setting will only take effect after parameter 90.10 ENC PAR REFRESH is used or after the JCU control unit is powered up the next time.
		EnDat or SSI encoder is used in continuous mode as encoder 2. [I.e. 90.02 ENCODER 2 SEL = (3) FEN-11 ABS and 91.02 ABS ENC INTERF = (2) EnDat or (4) SSI) and 91.30 ENDAT MODE = (1) Continuous (or 91.25 SSI MODE = (1) Continuous).]	If possible, use single position transfer instead of continuous position transfer (i.e. if encoder has incremental sin/cos signals): - Change parameter 91.25 SSI MODE / 91.30 ENDAT MODE to value (0) Initial pos.. Otherwise use EnDat/SSI encoder as encoder 1: - Change parameter 90.01 ENCODER 1 SEL to value (3) FEN-11 ABS and parameter 90.02 ENCODER 2 SEL to value (0) None. **Note:** The new setting will only take effect after parameter 90.10 ENC PAR REFRESH is used or after the JCU control unit is powered up the next time.

Code	Alarm (fieldbus code)	Cause	What to do
2026	ENC EMULATION FAILURE (0x7384)	Encoder emulation error	If position value used in emulation is measured by encoder: - Check that FEN-xx encoder used in emulation (90.03 EMUL MODE SEL) corresponds to FEN-xx encoder interface 1 or (and) 2 activated by parameter 90.01 ENCODER 1 SEL / 90.02 ENCODER 2 SEL. (Parameter 90.01/90.02 activates the position calculation of the used FEN-xx input). If position value used in emulation is determined by drive software: - Check that FEN-xx encoder used in emulation (90.03 EMUL MODE SEL) corresponds to FEN-xx encoder interface 1 or (and) 2 activated by parameter 90.01 ENCODER 1 SEL / 90.02 ENCODER 2 SEL (because position data used in emulation is written to FEN-xx during encoder data request). Encoder interface 2 is recommended. **Note:** The new setting will only take effect after parameter 90.10 ENC PAR REFRESH is used or after the JCU control unit is powered up the next time.
2027	FEN TEMP MEAS FAILURE (0x7385)	Error in temperature measurement when temperature sensor (KTY or PTC) connected to encoder interface FEN-xx is used.	Check that parameter 45.02 MOT TEMP SOURCE setting corresponds to encoder interface installation (9.20 OPTION SLOT 1 / 9.21 OPTION SLOT 2): If one FEN-xx module is used: - Parameter 45.02 MOT TEMP SOURCE setting must be either (2) KTY 1st FEN or (5) PTC 1st FEN. FEN-xx module can be in either Slot 1 or Slot 2. If two FEN-xx modules are used: - When parameter 45.02 MOT TEMP SOURCE setting is (2) KTY 1st FEN or (5) PTC 1st FEN, the encoder installed in drive Slot 1 is used. - When parameter 45.02 MOT TEMP SOURCE setting is (3) KTY 2nd FEN or (6) PTC 2nd FEN, the encoder installed in drive Slot 2 is used.
		Error in temperature measurement when KTY sensor connected to encoder interface FEN-01 is used.	FEN-01 does not support temperature measurement with KTY sensor. Use PTC sensor or other encoder interface module.

Code	Alarm (fieldbus code)	Cause	What to do
2028	ENC EMUL MAX FREQ (0x7386)	TTL pulse frequency used in encoder emulation.exceeds maximum allowed limit (500 kHz).	Decrease parameter 93.21 EMUL PULSE NR value. **Note:** The new setting will only take effect after parameter 90.10 ENC PAR REFRESH is used or after the JCU control unit is powered up the next time.
2029	ENC EMUL REF ERROR (0x7387)	Encoder emulation has failed due to failure in writing new (position) reference for emulation.	Contact your local ABB representative.
2030	RESOLVER AUTOTUNE ERR (0x7388)	Resolver autotuning routines, which are automatically started when resolver input is activated for the first time, have failed.	Check cable between resolver and resolver interface module (FEN-21) and order of connector signal wires at both ends of cable. Check resolver parameter settings. For resolver parameters and information, see parameter group 92 RESOLVER CONF on page 178. **Note:** Resolver autotuning routines should always be performed after resolver cable connection has been modified. Autotuning routines can be activated by setting parameter 92.02 EXC SIGNAL AMPL or 92.03 EXC SIGNAL FREQ, and then setting parameter 90.10 ENC PAR REFRESH to (1) Configure.
2031	ENCODER 1 CABLE (0x7389)	Encoder 1 cable fault detected.	Check cable between FEN-xx interface and encoder 1. After any modifications in cabling, re-configure interface by switching drive power off and on, or by activating parameter 90.10 ENC PAR REFRESH.
2032	ENCODER 2 CABLE (0x738A)	Encoder 2 cable fault detected.	Check cable between FEN-xx interface and encoder 2. After any modifications in cabling, re-configure interface by switching drive power off and on, or by activating parameter 90.10 ENC PAR REFRESH.
2033	D2D COMMUNICATION (0x7520) Programmable fault: 57.02 COMM LOSS FUNC	On the master drive: The drive has not been replied to by an activated follower for five consecutive polling cycles.	Check that all drives that are polled (parameters 57.04 and 57.05) on the drive-to-drive link are powered, properly connected to the link, and have the correct node address. Check the drive-to-drive link wiring.
		On a follower drive: The drive has not received new reference 1 and/or 2 for five consecutive reference handling cycles.	Check the settings of parameters 57.06 and 57.07 on the master drive. Check the drive-to-drive link wiring.

Code	Alarm (fieldbus code)	Cause	What to do
2034	D2D BUFFER OVERLOAD (0x7520) Programmable fault: 57.02 COMM LOSS FUNC	Transmission of drive-to-drive references failed because of message buffer overflow.	Contact your local ABB representative.
2035	PS COMM (0x5480)	Communication errors detected between the JCU Control Unit and the power unit of the drive.	Check the connections between the JCU Control Unit and the power unit.
2036	RESTORE (0x630D)	Restoration of backed-up parameters failed.	Contact your local ABB representative.
2037	CUR MEAS CALIBRATION (0x2280)	Current measurement calibration will occur at next start.	Informative alarm.
2038	AUTOPHASING (0x3187)	Autophasing will occur at next start.	Informative alarm.
2039	EARTH FAULT (0x2330) Programmable fault: 46.05 EARTH FAULT	Drive has detected load unbalance typically due to earth fault in motor or motor cable.	Check there are no power factor correction capacitors or surge absorbers in motor cable. Check that there is no earth fault in motor or motor cables: - measure insulation resistances of motor and motor cable. If no earth fault can be detected, contact your local ABB representative.
2041	MOTOR NOM VALUE (0x6383)	The motor configuration parameters are set incorrectly.	Check the settings of the motor configuration parameters in group 99 START-UP DATA.
		The drive is not dimensioned correctly.	Check that the drive is sized correctly for the motor.
2042	D2D CONFIG (0x7583)	The settings of drive-to-drive link configuration parameters (group 57) are incompatible.	Check the settings of the parameters in group 57 D2D COMMUNICATION.
2047	SPEED FEEDBACK (0x8480)	No speed feedback is received.	Check the settings of the parameters in group 22 SPEED FEEDBACK. Check encoder installation. See the description of fault 0039 (ENCODER1) for more information.
2048	OPTION COMM LOSS (0x7000)	Communication between drive and option module (FEN-xx and/or FIO-xx) is lost.	Check that option modules are properly connected to Slot 1 and (or) Slot 2. Check that option modules or Slot 1/2 connectors are not damaged. To determine whether module or connector is damaged: Test each module individually in Slot 1 and Slot 2.
2072	DC NOT CHARGED (0x3250)	The voltage of the intermediate DC circuit has not yet risen to operating level.	Wait for the DC voltage to rise.

Code	Alarm (fieldbus code)	Cause	What to do
2073	SPEED TUNE FAIL (0x8481)	Speed controller autotune routine did not finish successfully.	See parameter 28.16 PI TUNE MODE.
2075	LOW VOLT MOD CON (0xC015)	Low voltage mode has been activated but the parameter settings are outside allowable limits.	Check the Low voltage mode parameters in group 47 VOLTAGE CTRL. See also section *Low voltage mode* on page 43.

Fault messages generated by the drive

Code	Fault (fieldbus code)	Cause	What to do
0001	OVERCURRENT (0x2310)	Output current has exceeded internal fault limit.	Check motor load. Check acceleration time. See parameter group 25 SPEED REF RAMP on page 114. Check motor and motor cable (including phasing and delta/star connection). Check that the start-up data in parameter group 99 corresponds to the motor rating plate. Check that there are no power factor correction capacitors or surge absorbers in motor cable. Check encoder cable (including phasing).
0002	DC OVERVOLTAGE (0x3210)	Excessive intermediate circuit DC voltage.	Check that overvoltage controller is on, parameter 47.01 OVERVOLTAGE CTRL. Check mains for static or transient overvoltage. Check brake chopper and resistor (if used). Check deceleration time. Use coast-to-stop function (if applicable). Retrofit frequency converter with brake chopper and brake resistor.
0003	DEVICE OVERTEMP (0x4210)	Measured drive temperature has exceeded internal fault limit.	Check ambient conditions. Check air flow and fan operation. Check heatsink fins for dust pick-up. Check motor power against unit power.
0004	SHORT CIRCUIT (0x2340)	Short-circuit in motor cable(s) or motor.	Check motor and motor cable. Check there are no power factor correction capacitors or surge absorbers in motor cable.
0005	DC UNDERVOLTAGE (0x3220)	Intermediate circuit DC voltage is not sufficient due to missing mains phase, blown fuse or rectifier bridge internal fault.	Check mains supply and fuses.
0006	EARTH FAULT (0x2330) Programmable fault: 46.05 EARTH FAULT	Drive has detected load unbalance typically due to earth fault in motor or motor cable.	Check there are no power factor correction capacitors or surge absorbers in motor cable. Check that there is no earth fault in motor or motor cables: - measure insulation resistances of motor and motor cable. If no earth fault can be detected, contact your local ABB representative.
0007	FAN FAULT (0xFF83)	Fan is not able to rotate freely or fan is disconnected. Fan operation is monitored by measuring fan current.	Check fan operation and connection.

Code	Fault (fieldbus code)	Cause	What to do
0008	IGBT OVERTEMP (0x7184)	Drive temperature based on thermal model has exceeded internal fault limit.	Check ambient conditions. Check air flow and fan operation. Check heatsink fins for dust pick-up. Check motor power against unit power.
0009	BC WIRING (0x7111)	Brake resistor short circuit or brake chopper control fault.	Check brake chopper and brake resistor connection. Ensure brake resistor is not damaged.
0010	BC SHORT CIRCUIT (0x7113)	Short circuit in brake chopper IGBT.	Ensure brake resistor is connected and not damaged.
0011	BC OVERHEAT (0x7181)	Brake chopper IGBT temperature has exceeded internal fault limit.	Let chopper cool down. Check for excessive ambient temperature. Check for cooling fan failure. Check for obstructions in the air flow. Check the dimensioning and cooling of the cabinet. Check resistor overload protection function settings, parameters 48.03…48.05. Check that braking cycle meets allowed limits. Check that drive supply AC voltage is not excessive.
0012	BR OVERHEAT (0x7112)	Brake resistor temperature has exceeded fault limit defined by parameter 48.06 BR TEMP FAULTLIM.	Stop drive. Let resistor cool down. Check resistor overload protection function settings, parameters 48.01…48.05. Check fault limit setting, parameter 48.06. Check that braking cycle meets allowed limits.
0013	CURR MEAS GAIN (0x3183)	Difference between output phase U2 and W2 current measurement gain is too great.	Contact your local ABB representative.
0014	CABLE CROSS CON (0x3181) Programmable fault: 46.08 CROSS CONNECTION	Incorrect input power and motor cable connection (i.e. input power cable is connected to drive motor connection).	Check input power connections.
0015	SUPPLY PHASE (0x3130) Programmable fault: 46.06 SUPPL PHS LOSS	Intermediate circuit DC voltage is oscillating due to missing input power line phase or blown fuse.	Check input power line fuses. Check for input power supply imbalance.
0016	MOTOR PHASE (0x3182) Programmable fault: 46.04 MOT PHASE LOSS	Motor circuit fault due to missing motor connection (all three phases are not connected).	Connect motor cable.

Code	Fault (fieldbus code)	Cause	What to do
0017	ID-RUN FAULT (0xFF84)	Motor ID Run is not completed successfully.	Check the fault logger for a fault code extension. See appropriate actions for each extension below.
	Fault code extension: 1	The ID run cannot be completed because the maximum current setting and/or the internal current limit of the drive is too low.	Check setting of parameters 99.06 MOT NOM CURRENT and 20.05 MAXIMUM CURRENT. Make sure that 20.05 MAXIMUM CURRENT ≥ 99.06 MOT NOM CURRENT. Check that the drive is dimensioned correctly according to the motor.
	Fault code extension: 2	The ID run cannot be completed because the maximum speed setting and/or calculated field weakening point is too low.	Check setting of parameters 99.07 MOT NOM VOLTAGE, 99.08 MOT NOM FREQ, 99.09 MOT NOM SPEED, 20.01 MAXIMUM SPEED and 20.02 MINIMUM SPEED. Make sure that • 20.01 MAXIMUM SPEED > (0.55 × 99.09 MOT NOM SPEED) > (0.50 × synchronous speed), • 20.02 MINIMUM SPEED ≤ 0, and • supply voltage ≥ (0.66 × 99.07 MOT NOM VOLTAGE).
	Fault code extension: 3	The ID run cannot be completed because the maximum torque setting is too low.	Check setting of parameters 99.12 MOT NOM TORQUE and 20.06 MAXIMUM TORQUE. Make sure that 20.06 MAXIMUM TORQUE ≥ 100%.
	Fault code extension: 4…16	Internal error.	Contact your local ABB representative.
0018	CURR U2 MEAS (0x3184)	Measured offset error of U2 output phase current measurement is too great. (Offset value is updated during current calibration.)	Contact your local ABB representative.
0019	CURR V2 MEAS (0x3185)	Measured offset error of V2 output phase current measurement is too great. (Offset value is updated during current calibration.)	Contact your local ABB representative.
0020	CURR W2 MEAS (0x3186)	Measured offset error of W2 output phase current measurement is too great. (Offset value is updated during current calibration.)	Contact your local ABB representative.
0021	STO1 LOST (0x8182)	Safe Torque Off function is active, i.e. safety circuit signal 1 connected between X6:1 and X6:3 is lost while drive is at stopped state and parameter 46.07 STO DIAGNOSTIC setting is (2) Alarm or (3) No.	Check safety circuit connections. For more information, see appropriate drive hardware manual.

Code	Fault (fieldbus code)	Cause	What to do
0022	STO2 LOST (0x8183)	Safe Torque Off function is active, i.e. safety circuit signal 2 connected between X6:2 and X6:4 is lost while drive is at stopped state and parameter 46.07 STO DIAGNOSTIC setting is (2) Alarm or (3) No.	Check safety circuit connections. For more information, see appropriate drive hardware manual.
0024	INTBOARD OVERTEMP (0x7182)	Interface board (between power unit and control unit) temperature has exceeded internal fault limit.	Let drive cool down. Check for excessive ambient temperature. Check for cooling fan failure. Check for obstructions in the air flow. Check the dimensioning and cooling of the cabinet.
0025	BC MOD OVERTEMP (0x7183)	Input bridge or brake chopper temperature has exceeded internal fault limit.	Let drive cool down. Check for excessive ambient temperature. Check for cooling fan failure. Check for obstructions in the air flow. Check the dimensioning and cooling of the cabinet.
0026	AUTOPHASING (0x3187)	Autophasing routine (see section *Autophasing* on page 38) failed.	Try other autophasing modes (see parameter 11.07 AUTOPHASING MODE) if possible.
0027	PU LOST (0x5400)	Connection between the JCU Control Unit and the power unit of the drive is lost.	Check the connections between the JCU Control Unit and the power unit.
0028	PS COMM (0x5480)	Communication errors detected between the JCU Control Unit and the power unit of the drive.	Check the connections between the JCU Control Unit and the power unit.
0029	IN CHOKE TEMP (0xFF81)	Temperature of internal AC choke excessive.	Check cooling fan.
0030	EXTERNAL (0x9000)	Fault in external device. (This information is configured through one of programmable digital inputs.)	Check external devices for faults. Check setting of parameter 46.01 EXTERNAL FAULT.
0031	SAFE TORQUE OFF (0xFF7A) Programmable fault: 46.07 STO DIAGNOSTIC	Safe Torque Off function is active, i.e. safety circuit signal(s) connected to connector X6 is lost - during drive start or drive run or - while drive is stopped and parameter 46.07 STO DIAGNOSTIC setting is (1) Fault.	Check safety circuit connections. For more information, see appropriate drive hardware manual.

Code	Fault (fieldbus code)	Cause	What to do
0032	OVERSPEED (0x7310)	Motor is turning faster than highest allowed speed due to incorrectly set minimum/ maximum speed, insufficient braking torque or changes in load when using torque reference.	Check minimum/maximum speed settings, parameters 20.01 MAXIMUM SPEED and 20.02 MINIMUM SPEED. Check adequacy of motor braking torque. Check applicability of torque control. Check need for brake chopper and resistor(s).
0033	BRAKE START TORQUE (0x7185) Programmable fault: 35.09 BRAKE FAULT FUNC	Mechanical brake fault. Fault is activated if required motor starting torque, 35.06 BRAKE OPEN TORQ, is not achieved.	Check brake open torque setting, parameter 35.06. Check drive torque and current limits. See parameter group 20 LIMITS on page 102.
0034	BRAKE NOT CLOSED (0x7186) Programmable fault: 35.09 BRAKE FAULT FUNC	Mechanical brake control fault. Fault is activated if brake acknowledgement is not as expected during brake closing.	Check mechanical brake connection. Check mechanical brake settings, parameters 35.01…35.09. To determine whether problem is with acknowledgement signal or brake: Check if brake is closed or open.
0035	BRAKE NOT OPEN (0x7187) Programmable fault: 35.09 BRAKE FAULT FUNC	Mechanical brake control fault. Fault is activated if brake acknowledgement is not as expected during brake opening.	Check mechanical brake connection. Check mechanical brake settings, parameters 35.01…35.08. To determine whether problem is with acknowledgement signal or brake: Check if brake is closed or open.
0036	LOCAL CTRL LOSS (0x5300) Programmable fault: 46.03 LOCAL CTRL LOSS	Control panel or PC tool selected as active control location for drive has ceased communicating.	Check PC tool or control panel connection. Check control panel connector. Replace control panel in mounting platform.
0037	NVMEMCORRUPTED (0x6320)	Drive internal fault **Note:** This fault cannot be reset.	Check the fault logger for a fault code extension. See appropriate actions for each extension below.
	Fault code extension: 2051	Total number of parameters (including unused space between parameters) exceeds firmware maximum.	Move parameters from the firmware groups to the application groups. Reduce the number of parameters.
	Fault code extension: Other	Drive internal fault.	Contact your local ABB representative.
0038	OPTION COMM LOSS (0x7000)	Communication between drive and option module (FEN-xx and/or FIO-xx) is lost.	Check that option modules are properly connected to Slot 1 and (or) Slot 2. Check that option modules or Slot 1/2 connectors are not damaged. To determine whether module or connector is damaged: Test each module individually in Slot 1 and Slot 2.

Code	Fault (fieldbus code)	Cause	What to do
0039	ENCODER1 (0x7301)	Encoder 1 feedback fault	If fault appears during first start-up before encoder feedback is used: - Check cable between encoder and encoder interface module (FEN-xx) and order of connector signal wires at both ends of cable. If absolute encoder, EnDat/Hiperface/SSI, with incremental sin/cos pulses is used, incorrect wiring can be located as follows: Disable serial link (zero position) by setting parameter 91.02 ABS ENC INTERF to (0) None and test encoder operation: - If encoder fault is not activated, check serial link data wiring. Note that zero position is not taken into account when serial link is disabled. - If encoder fault is activated, check serial link and sin/cos signal wiring. **Note:** Because only zero position is requested through serial link and during run, position is updated according to sin/cos pulses. - Check encoder parameter settings. If fault appears after encoder feedback has already been used or during drive run: - Check that encoder connection wiring or encoder is not damaged. - Check that encoder interface module (FEN-xx) connection or module is not damaged. - Check earthings (when disturbances are detected in communication between encoder interface module and encoder). For more information on encoders, see parameter groups 90 ENC MODULE SEL (page 169), 91 ABSOL ENC CONF (page 173), 92 RESOLVER CONF (page 178) and 93 PULSE ENC CONF (page 179).

Code	Fault (fieldbus code)	Cause	What to do
0040	ENCODER2 (0x7381)	Encoder 2 feedback fault	See fault ENCODER1.
		EnDat or SSI encoder is used in continuous mode as encoder 2. [I.e. 90.02 ENCODER 2 SEL = (3) FEN-11 ABS and 91.02 ABS ENC INTERF = (2) EnDat or (4) SSI and 91.30 ENDAT MODE = (1) Continuous (or 91.25 SSI MODE = (1) Continuous).]	If possible, use single position transfer instead of continuous position transfer (i.e. if encoder has incremental sin/cos signals): - Change parameter 91.25 SSI MODE / 91.30 ENDAT MODE to value (0) Initial pos.. Otherwise use Endat/SSI encoder as encoder 1: - Change parameter 90.01 ENCODER 1 SEL to value (3) FEN-11 ABS and parameter 90.02 ENCODER 2 SEL to value (0) None. **Note:** The new setting will only take effect after parameter 90.10 ENC PAR REFRESH is used or after the JCU control unit is powered up the next time.
0045	FIELDBUS COMM (0x7510) Programmable fault: 50.02 COMM LOSS FUNC	Cyclical communication between drive and fieldbus adapter module or between PLC and fieldbus adapter module is lost.	Check status of fieldbus communication. See appropriate User's Manual of fieldbus adapter module. Check fieldbus parameter settings. See parameter group 50 FIELDBUS on page 156. Check cable connections. Check if communication master can communicate.
0046	FB MAPPING FILE (0x6306)	Drive internal fault	Contact your local ABB representative.
0047	MOTOR OVERTEMP (0x4310) Programmable fault: 45.01 MOT TEMP PROT	Estimated motor temperature (based on motor thermal model) has exceeded fault limit defined by parameter 45.04 MOT TEMP FLT LIM.	Check motor ratings and load. Let motor cool down. Ensure proper motor cooling: Check cooling fan, clean cooling surfaces, etc. Check value of fault limit. Check motor thermal model settings, parameters 45.06…45.08 and 45.10 MOT THERM TIME.
		Measured motor temperature has exceeded fault limit defined by parameter 45.04 MOT TEMP FLT LIM.	Check that actual number of sensors corresponds to value set by parameter 45.02 MOT TEMP SOURCE. Check motor ratings and load. Let motor cool down. Ensure proper motor cooling: Check cooling fan, clean cooling surfaces, etc. Check value of fault limit.
0049	AI SUPERVISION (0x8110) Programmable fault: 13.12 AI SUPERVISION	Analogue input AI1 or AI2 signal has reached limit defined by parameter 13.13 AI SUPERVIS ACT.	Check analogue input AI1/2 source and connections. Check analogue input AI1/2 minimum and maximum limit settings, parameters 13.02 and 13.03 / 13.07 and 13.08.

Code	Fault (fieldbus code)	Cause	What to do
0050	ENCODER 1 CABLE (0x7389) Programmable fault: 90.05 ENC CABLE FAULT	Encoder 1 cable fault detected.	Check cable between FEN-xx interface and encoder 1. After any modifications in cabling, re-configure interface by switching drive power off and on, or by activating parameter 90.10 ENC PAR REFRESH.
0051	ENCODER 2 CABLE (0x738A) Programmable fault: 90.05 ENC CABLE FAULT	Encoder 2 cable fault detected.	Check cable between FEN-xx interface and encoder 2. After any modifications in cabling, re-configure interface by switching drive power off and on, or by activating parameter 90.10 ENC PAR REFRESH.
0052	D2D CONFIG (0x7583)	Configuration of the drive-to-drive link has failed for a reason other than those indicated by alarm 2042, for example start inhibition is requested but not granted.	Contact your local ABB representative.
0053	D2D COMM (0x7520) Programmable fault: 57.02 COMM LOSS FUNC	On the master drive: The drive has not been replied to by an activated follower for five consecutive polling cycles.	Check that all drives that are polled (parameters 57.04 FOLLOWER MASK 1 and 57.05 FOLLOWER MASK 2) on the drive-to-drive link are powered, properly connected to the link, and have the correct node address. Check the drive-to-drive link wiring.
		On a follower drive: The drive has not received new reference 1 and/or 2 for five consecutive reference handling cycles.	Check the settings of parameters 57.06 REF 1 SRC and 57.07 REF 2 SRC on the master drive. Check the drive-to-drive link wiring.
0054	D2D BUF OVLOAD (0x7520) Programmable fault: 57.02 COMM LOSS FUNC	Transmission of drive-to-drive references failed because of message buffer overflow.	Contact your local ABB representative.
0055	TECH LIB (0x6382)	Resettable fault generated by a technology library.	Refer to the documentation of the technology library.
0056	TECH LIB CRITICAL (0x6382)	Permanent fault generated by a technology library.	Refer to the documentation of the technology library.
0057	FORCED TRIP (0xFF90)	Generic Drive Communication Profile trip command.	Check PLC status.
0058	FIELDBUS PAR ERROR (0x6320)	The drive does not have a functionality requested by PLC, or requested functionality has not been activated.	Check PLC programming. Check fieldbus parameter settings. See parameter group 50 FIELDBUS on page 156.
0061	SPEED FEEDBACK (0x8480)	No speed feedback is received.	Check the settings of the parameters in group 22 SPEED FEEDBACK. Check encoder installation. See the description of fault 0039 (ENCODER1) for more information.

Code	Fault (fieldbus code)	Cause	What to do
0062	D2D SLOT COMM (0x7584)	Drive-to-drive link is set to use an FMBA module for communication, but no module is detected in specified slot.	Check the settings of parameters 57.01 LINK MODE and 57.15 D2D COMM PORT. Ensure that the FMBA module has been detected by checking parameters 9.20…9.22. Check that the FMBA module is correctly wired. Try installing the FMBA module into another slot. If the problem persists, contact your local ABB representative.
0067	FPGA ERROR1 (0x5401)	Drive internal fault	Contact your local ABB representative.
0068	FPGA ERROR2 (0x5402)	Drive internal fault	Contact your local ABB representative.
0069	ADC ERROR (0x5403)	Drive internal fault	Contact your local ABB representative.
0201	T2 OVERLOAD (0x0201)	Firmware time level 2 overload **Note:** This fault cannot be reset.	Contact your local ABB representative.
0202	T3 OVERLOAD (0x6100)	Firmware time level 3 overload **Note:** This fault cannot be reset.	Contact your local ABB representative.
0203	T4 OVERLOAD (0x6100)	Firmware time level 4 overload **Note:** This fault cannot be reset.	Contact your local ABB representative.
0204	T5 OVERLOAD (0x6100)	Firmware time level 5 overload **Note:** This fault cannot be reset.	Contact your local ABB representative.
0205	A1 OVERLOAD (0x6100)	Application time level 1 fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0206	A2 OVERLOAD (0x6100)	Application time level 2 fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0207	A1 INIT FAULT (0x6100)	Application task creation fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0208	A2 INIT FAULT (0x6100)	Application task creation fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0209	STACK ERROR (0x6100)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.

Code	Fault (fieldbus code)	Cause	What to do
0210	JMU MISSING (0xFF61)	JMU Memory Unit is missing or broken.	Check that the JMU is properly installed. If the problem persists, replace JMU.
0301	UFF FILE READ (0x6300)	File read error **Note:** This fault cannot be reset.	Contact your local ABB representative.
0302	APPL DIR CREATION (0x6100)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0303	FPGA CONFIG DIR (0x6100)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0304	PU RATING ID (0x5483)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0305	RATING DATABASE (0x6100)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0306	LICENSING (0x6100)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0307	DEFAULT FILE (0x6100)	Drive internal fault **Note:** This fault cannot be reset.	Contact your local ABB representative.
0308	APPL FILE PAR CONF (0x6300)	Corrupted application file **Note:** This fault cannot be reset.	Reload application. If fault is still active, contact your local ABB representative.
0309	APPL LOADING (0x6300)	Application file incompatible or corrupted. **Note:** This fault cannot be reset.	Check the fault logger for a fault code extension. See appropriate actions for each extension below.
	Fault code extension: 8	Template used in the application incompatible with drive firmware.	Change the template of the application in DriveSPC.
	Fault code extension: 10	Parameters defined in the application conflict with existing drive parameters.	Check the application for conflicting parameters.
	Fault code extension: 35	Application memory full.	Contact your local ABB representative.
	Fault code extension: Other	Corrupted application file	Reload application. If fault is still active, contact your local ABB representative.

Code	Fault (fieldbus code)	Cause	What to do
0310	USERSET LOAD (0xFF69)	Loading of user set is not successfully completed because: - requested user set does not exist - user set is not compatible with drive program - drive has been switched off during loading.	Reload.
0311	USERSET SAVE (0xFF69)	User set is not saved because of memory corruption.	Check the setting of parameter 95.01 CTRL UNIT SUPPLY. If the fault still occurs, contact your local ABB representative.
0312	UFF OVERSIZE (0x6300)	UFF file is too big.	Contact your local ABB representative.
0313	UFF EOF (0x6300)	UFF file structure failure	Contact your local ABB representative.
0314	TECH LIB INTERFACE (0x6100)	Incompatible firmware interface **Note:** This fault cannot be reset.	Contact your local ABB representative.
0315	RESTORE FILE (0x630D)	Restoration of backed-up parameters failed.	Contact your local ABB representative.
0316	DAPS MISMATCH (0x5484)	Mismatch between JCU Control Unit firmware and power unit logic versions.	Contact your local ABB representative.
0317	SOLUTION FAULT (0x6200)	Fault generated by function block SOLUTION_FAULT in the application program.	Check the usage of the SOLUTION_FAULT block in the application program.

Standard function blocks

What this chapter contains

This chapter describes the standard function blocks. The blocks are grouped according to the grouping in the DriveSPC tool.

The number in brackets in the standard block heading is the block number.

Note: The given execution times can vary depending on the used drive application.

Terms

Data type	Description	Range
Boolean	Boolean	0 or 1
DINT	32-bit integer value (31 bits + sign)	-2147483648…2147483647
INT	16-bit integer value (15 bits + sign)	-32768…32767
PB	Packed Boolean	0 or 1 for each individual bit
REAL	16-bit value 16-bit value (31 bits + sign) = integer value = fractional value	-32768,99998…32767,9998
REAL24	8-bit value 24-bit value (31 bits + sign) = integer value = fractional value	-128,0…127,999

Alphabetical index

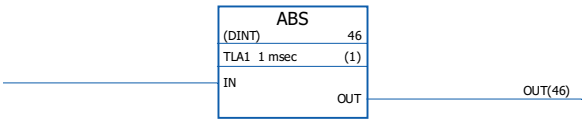
ABS 233
ADD 233
AND 237
BGET 242
BITAND 242
BITOR 243
BOOL_TO_DINT 254
BOOL_TO_INT 255
BOP 301
BSET 243
CRITSPEED 284
CTD 263
CTD_DINT 263
CTU 264
CTU_DINT 265
CTUD 266
CTUD_DINT 268
CYCLET 285
D2D_Conf 246
D2D_McastToken 247
D2D_SendMessage 247
DATA CONTAINER 285
DEMUX-I 306
DEMUX-MI 306
DINT_TO_BOOL 256
DINT_TO_INT 257
DINT_TO_REALn 257
DINT_TO_REALn_SIMP 258
DIV 233
DS_ReadLocal 249
DS_WriteLocal 250
ELSE 301
ELSEIF 301
ENDIF 302
EQ 251
EXPT 234
FILT1 295
FILT2 295
FIO_01_slot1 273
FIO_01_slot2 274
FIO_11_AI_slot1 275
FIO_11_AI_slot2 277
FIO_11_AO_slot1 279
FIO_11_AO_slot2 280
FIO_11_DIO_slot1 282
FIO_11_DIO_slot2 282
FTRIG 270
FUNG-1V 286
GE 251
GetBitPtr 298
GetValPtr 298
GT 251
IF 302
INT 287
INT_TO_BOOL 259
INT_TO_DINT 259
LE 252
LEAD/LAG 297
LIMIT 304
LT 252
MAX 304
MIN 304
MOD 234
MONO 309
MOTPOT 288
MOVE 235
MUL 235
MULDIV 235
MUX 305
NE 253
NOT 237
OR 238
PARRD 298
PARRDINTR 299
PARRDPTR 299
PARWR 300
PID 289
RAMP 291
REAL_TO_REAL24 260
REAL24_TO_REAL 260
REALn_TO_DINT 261
REALn_TO_DINT_SIMP 261
REG 244
REG-G 292
ROL 238
ROR 239
RS 270
RTRIG 271
SEL 305
SHL 239
SHR 240
SOLUTION_FAULT 294
SQRT 236
SR 272
SR-D 245
SUB 236
SWITCH 307
SWITCHC 308
TOF 310
TON 310
TP 311
XOR 240

Arithmetic

ABS

(10001)

Illustration



Execution time 0.53 µs

Operation The output (OUT) is the absolute value of the input (IN).

OUT = | IN |

Inputs The input data type is selected by the user.

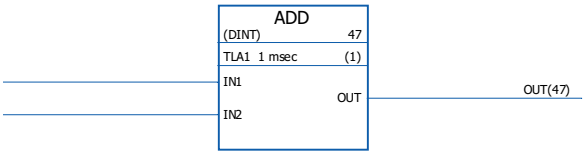
Input (IN): DINT, INT, REAL or REAL24

Outputs Output (OUT): DINT, INT, REAL or REAL24

ADD

(10000)

Illustration



Execution time 3.36 µs (when two inputs are used) + 0.52 µs (for every additional input). When all inputs are used, the execution time is 18.87 µs.

Operation The output (OUT) is the sum of the inputs (IN1…IN32).

OUT = IN1 + IN2 + … + IN32

The output value is limited to the maximum and minimum values defined by the selected data type range.

Inputs The input data type and the number of the inputs (2…32) are selected by the user.

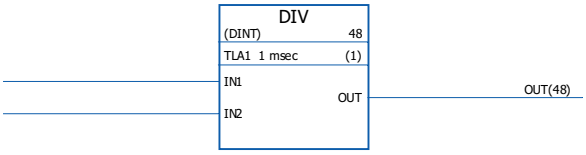
Input (IN1…IN32): DINT, INT, REAL or REAL24

Outputs Output (OUT): DINT, INT, REAL or REAL24

DIV

(10002)

Illustration



Execution time 2.55 µs

Operation The output (OUT) is input IN1 divided by input IN2.

OUT = IN1/IN2

The output value is limited to the maximum and minimum values defined by the selected data type range.

If the divider (IN2) is 0, the output is 0.

Inputs The input data type is selected by the user.

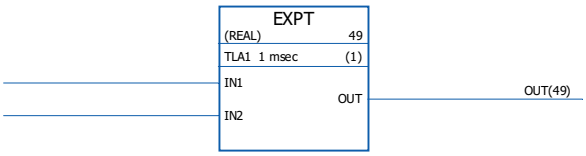
Input (IN1, IN2): INT, DINT, REAL, REAL24

Outputs Output (OUT): INT, DINT, REAL, REAL24

EXPT

(10003)

Illustration



Execution time 81.90 µs

Operation The output (OUT) is input IN1 raised to the power of the input IN2:

OUT = $IN1^{IN2}$

If input IN1 is 0, the output is 0.

The output value is limited to the maximum value defined by the selected data type range.

Note: The execution of the EXPT function is slow.

Inputs The input data type is selected by the user.

Input (IN1): REAL, REAL24

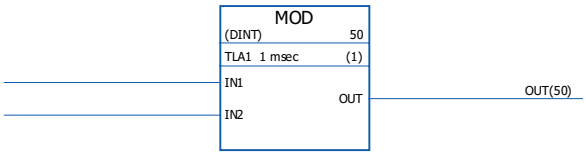
Input (IN2): REAL

Outputs Output (OUT): REAL, REAL24

MOD

(10004)

Illustration



Execution time 1.67 µs

Operation The output (OUT) is the remainder of the division of the inputs IN1 and IN2.

OUT = remainder of IN1/IN2

If input IN2 is zero, the output is zero.

Inputs The input data type is selected by the user.

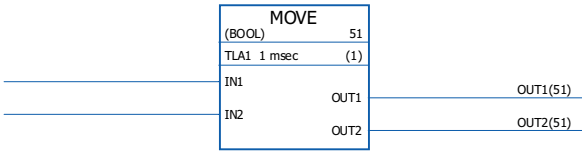
Input (IN1, IN2): INT, DINT

Outputs Output (OUT): INT, DINT

MOVE

(10005)

Illustration



Execution time 2.10 µs (when two inputs are used) + 0.42 µs (for every additional input). When all inputs are used, the execution time is 14.55 µs.

Operation Copies the input values (IN1…32) to the corresponding outputs (OUT1…32).

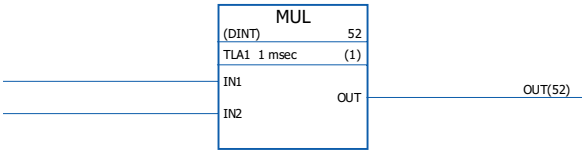
Inputs The input data type and number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24, Boolean

Outputs Output (OUT1…OUT32): INT, DINT, REAL, REAL24, Boolean

MUL

(10006)

Illustration



Execution time 3.47 µs (when two inputs are used) + 2.28 µs (for every additional input). When all inputs are used, the execution time is 71.73 µs.

Operation The output (OUT) is the product of the inputs (IN).
O = IN1 × IN2 × … × IN32
The output value is limited to the maximum and minimum values defined by the selected data type range.

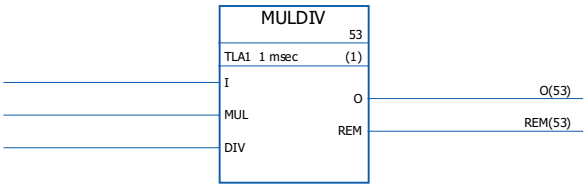
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): INT, DINT, REAL, REAL24

MULDIV

(10007)

Illustration



Execution time 7.10 µs

Operation The output (O) is the product of input IN and input MUL divided by input DIV.
Output = (I × MUL) / DIV
O = whole value. REM = remainder value.
Example: I = 2, MUL = 16 and DIV = 10:
(2 × 16) / 10 = 3.2, i.e. O = 3 and REM = 2
The output value is limited to the maximum and minimum values defined by the data type range.

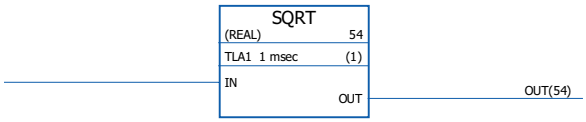
Inputs Input (I): DINT
Multiplier input (MUL): DINT
Divider input (DIV): DINT

Outputs Output (O): DINT
Remainder output (REM): DINT

SQRT

(10008)

Illustration



Execution time 2.09 µs

Operation Output (OUT) is the square root of the input (IN).
OUT = sqrt(IN)
Output is 0 if the input value is negative.

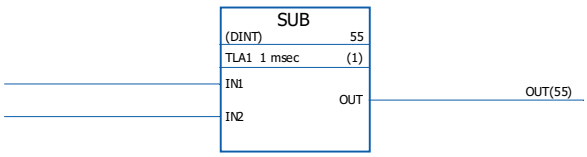
Inputs The input data type is selected by the user.
Input (IN): REAL, REAL24

Outputs Output (OUT): REAL, REAL24

SUB -

(10009)

Illustration



Execution time 2.33 µs

Operation Output (OUT) is the difference between the input signals (IN):
OUT = IN1 - IN2
The output value is limited to the maximum and minimum values defined by the selected data type range.

Inputs The input data type is selected by the user.
Input (IN1, IN2): INT, DINT, REAL, REAL24

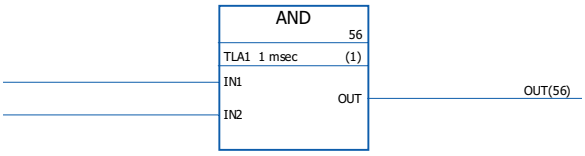
Outputs Output (OUT): INT, DINT, REAL, REAL24

Bitstring

AND

(10010)

Illustration



Execution time 1.55 µs (when two inputs are used) + 0.60 µs (for every additional input). When all inputs are used, the execution time is 19.55 µs.

Operation The output (OUT) is 1 if all the connected inputs (IN1…IN32) are 1. Otherwise the output is 0.

Truth table:

IN1	IN2	OUT
0	0	0
0	1	0
1	0	0
1	1	1

The inputs can be inverted.

Inputs The number of inputs is selected by the user.

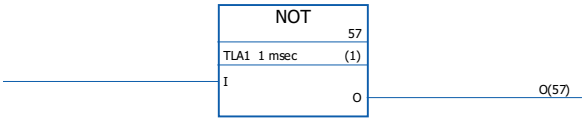
Input (IN1…IN32): Boolean

Outputs Output (OUT): Boolean

NOT

(10011)

Illustration



Execution time 0.32 µs

Operation The output (O) is 1 if the input (I) is 0. The output is 0 if the input is 1.

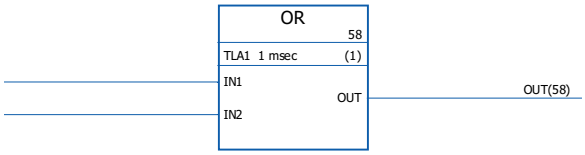
Inputs Input (I): Boolean

Outputs Output (O): Boolean

OR

(10012)

Illustration



Execution time 1.55 µs (when two inputs are used) + 0.60 µs (for every additional input). When all inputs are used, the execution time is 19.55 µs.

Operation The output (OUT) is 0, if all connected inputs (IN) are 0. Otherwise the output is 1.

Truth table:

IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	1

The inputs can be inverted.

Inputs The number of inputs (2…32) is selected by the user.

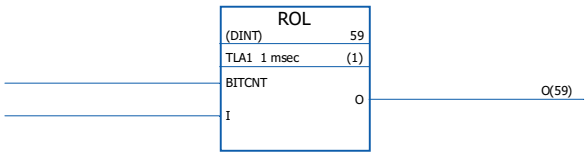
Input (IN1…IN32): Boolean

Outputs Output (OUT): Boolean

ROL

(10013)

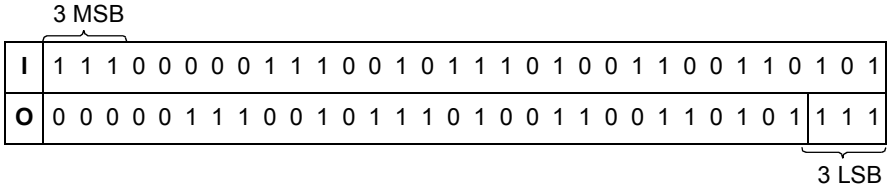
Illustration



Execution time 1.28 µs

Operation Input bits (I) are rotated to the left by the number (N) of bits defined by BITCNT. The N most significant bits (MSB) of the input are stored as the N least significant bits (LSB) of the output.

Example: If BITCNT = 3



Inputs The input data type is selected by the user.

Number of bits input (BITCNT): INT, DINT

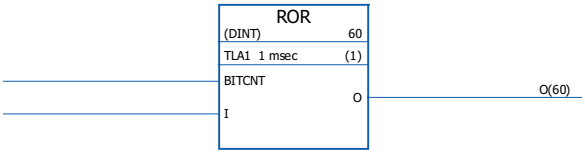
Input (I): INT, DINT

Outputs Output (O): INT, DINT

ROR

(10014)

Illustration



Execution time 1.28 µs

Operation Input bits (I) are rotated to the right by the number (N) of bits defined by BITCNT. The N least significant bits (LSB) of the input are stored as the N most significant bits (MSB) of the output.

Example: If BITCNT = 3

																											3 LSB			
I	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1
O	1	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0
	3 MSB																															

Inputs The input data type is selected by the user.

Number of bits input (BITCNT): INT, DINT

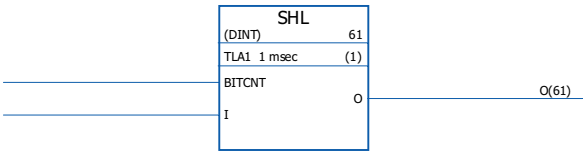
Input (I): INT, DINT

Outputs Output (O): INT, DINT

SHL

(10015)

Illustration



Execution time 0.80 µs

Operation Input bits (I) are rotated to the left by the number (N) of bits defined by BITCNT. The N most significant bits (MSB) of the input are lost and the N least significant bits (LSB) of the output are set to 0.

Example: If BITCNT = 3

	3 MSB																													
I	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1
O	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1	0	0	0
																													3 LSB			

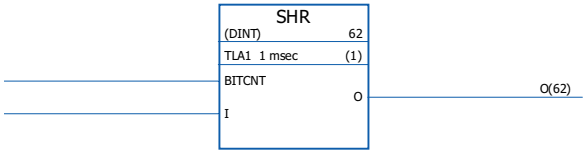
Inputs The input data type is selected by the user.
Number of bits (BITCNT): INT; DINT
Input (I): INT, DINT

Outputs Output (O): INT; DINT

SHR

(10016)

Illustration



Execution time 0.80 µs

Operation Input bits (I) are rotated to the right by the number (N) of bits defined by BITCNT. The N least significant bits (LSB) of the input are lost and the N most significant bits (MSB) of the output are set to 0.
Example: If BITCNT = 3

3 LSB

I	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1
O	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0

3 MSB

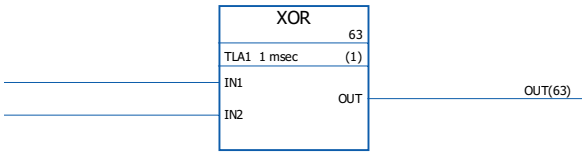
Inputs The input data type is selected by the user.
Number of bits (BITCNT): INT; DINT
Input (I): INT, DINT

Outputs Output (O): INT; DINT

XOR

(10017)

Illustration



Execution time 1.24 µs (when two inputs are used) + 0.72 µs (for every additional input). When all inputs are used, the execution time is 22.85 µs.

Operation

The output (OUT) is 1 if one of the connected inputs (IN1…IN32) is 1. Output is zero if all the inputs have the same value.

Example:

IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	0

The inputs can be inverted.

Inputs

The number of inputs (2…32) is selected by the user.

Input (IN1…IN32): Boolean

Outputs

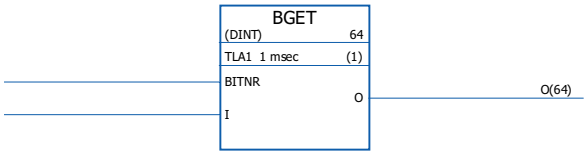
Output (OUT): Boolean

Bitwise

BGET

(10034)

Illustration



Execution time 0.88 µs

Operation The output (O) is the value of the selected bit (BITNR) of the input (I).
BITNR: Bit number (0 = bit number 0, 31 = bit number 31)
If bit number is not in the range of 0…31 (for DINT) or 0…15 (for INT), the output is 0.

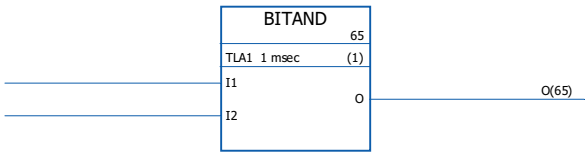
Inputs The input data type is selected by the user.
Number of the bit (BITNR): DINT
Input (I): DINT, INT

Outputs Output (O): Boolean

BITAND

(10035)

Illustration



Execution time 0.32 µs

Operation The output (O) bit value is 1 if the corresponding bit values of the inputs (I1 and I2) are 1. Otherwise the output bit value is 0.
Example:

I1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1
I2	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1	1	1	1
O	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	1

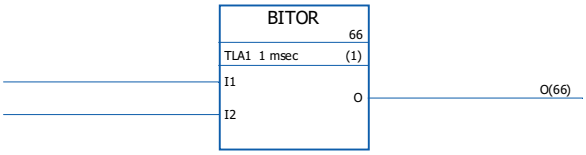
Inputs Input (I1, I2): DINT

Outputs Output (O): DINT

BITOR

(10036)

Illustration



Execution time 0.32 µs

Operation The output (O) bit value is 1 if the corresponding bit value of any of the inputs (I1 or I2) is 1. Otherwise the output bit value is 0.

Example:

I1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1
I2	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1	1	1	1
O	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	1

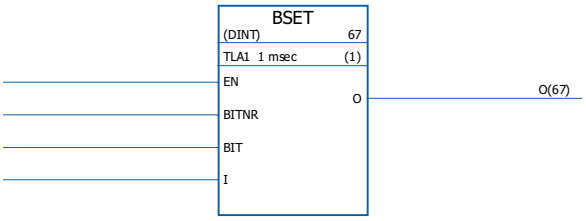
Input Input (I1, I2): DINT

Output Output (O): DINT

BSET

(10037)

Illustration



Execution time 1.36 µs

Operation The value of a selected bit (BITNR) of the input (I) is set as defined by the bit value input (BIT). The function must be enabled by the enable input (EN).

BITNR: Bit number (0 = bit number 0, 31 = bit number 31)

If BITNR is not in the range of 0…31 (for DINT) or 0…15 (for INT) or if EN is reset to zero, the input value is stored to the output as it is (i.e. no bit setting occurs).

Example:

EN = 1, BITNR = 3, BIT = 0

IN = 0000 0000 1111 1111

O = 0000 0000 1111 0111

Inputs The input data type is selected by the user.

Enable input (EN): Boolean

Number of the bit (BITNR): DINT

Bit value input (BIT): Boolean

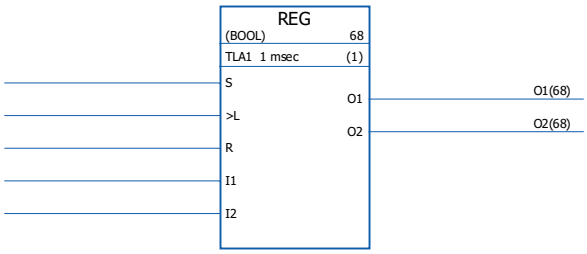
Input (I): INT, DINT

Outputs Output (O): INT, DINT

REG

(10038)

Illustration



Execution time 2.27 µs (when two inputs are used) + 1.02 µs (for every additional input). When all inputs are used, the execution time is 32.87 µs.

Operation The input (I1…I32) value is stored to the corresponding output (O1…O32) if the load input (L) is set to 1 or the set input (S) is 1. When the load input is set to 1, the input value is stored to the output only once. When the set input is 1, the input value is stored to the output every time the block is executed. The set input overrides the load input.

If the reset input (R) is 1, all connected outputs are 0.

Example:

S	R	L	I	$O1_{previous}$	O1
0	0	0	10	15	15
0	0	0->1	20	15	20
0	1	0	30	20	0
0	1	0->1	40	0	0
1	0	0	50	0	50
1	0	0->1	60	50	60
1	1	0	70	60	0
1	1	0->1	80	0	0
$O1_{previous}$ is the previous cycle output value.					

Inputs The input data type and number of inputs (1…32) are selected by the user.

Set input (S): Boolean

Load input (L): Boolean

Reset input (R): Boolean

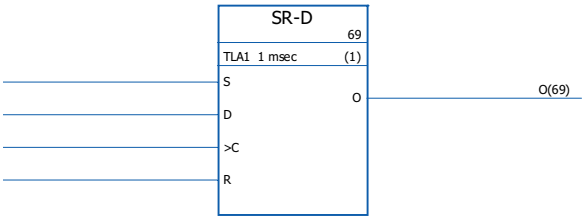
Input (I1…I32): Boolean, INT, DINT, REAL, REAL24

Outputs Output (O1…O32): Boolean, INT, DINT, REAL, REAL24

SR-D

(10039)

Illustration



Execution time 1.04 µs

Operation

When clock input (C) is set to 1, the data input (D) value is stored to the output (O). When reset input (R) is set to 1, the output is set to 0.

If only set (S) and reset (R) inputs are used, SR-D block acts as an SR block:
The output is 1 if the set input (S) is 1. The output will retain the previous output state if the set input (S) and reset input (R) are 0. The output is 0 if the set input is 0 and the reset input is 1.

Truth table:

S	R	D	C	$O_{previous}$	O
0	0	0	0	0	0 (= Previous output value)
0	0	0	0 -> 1	0	0 (= Data input value)
0	0	1	0	0	0 (= Previous output value)
0	0	1	0 -> 1	0	1 (= Data input value)
0	1	0	0	1	0 (Reset)
0	1	0	0 -> 1	0	0 (Reset)
0	1	1	0	0	0 (Reset)
0	1	1	0 -> 1	0	0 (Reset)
1	0	0	0	0	1 (= Set value)
1	0	0	0 -> 1	1	0 (= Data input value) for one execution cycle, then changes to 1 according to the set input (S = 1).
1	0	1	0	1	1 (= Set value)
1	0	1	0 -> 1	1	1 (= Data input value)
1	1	0	0	1	0 (Reset)
1	1	0	0 -> 1	0	0 (Reset)
1	1	1	0	0	0 (Reset)
1	1	1	0 -> 1	0	0 (Reset)
$O_{previous}$ is the previous cycle output value.					

Inputs

Set input (S): Boolean

Data input (D): Boolean

Clock input (C): Boolean

Reset input (R): Boolean

Outputs

Output (O): Boolean

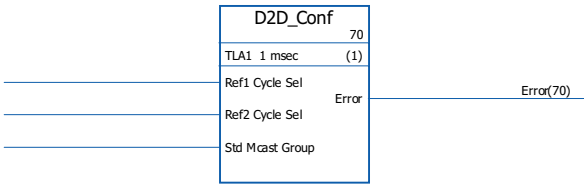
Communication

See also *Appendix B – Drive-to-drive link* (page 331).

D2D_Conf

(10092)

Illustration



Execution time -

Operation Defines handling interval for drive-to-drive references 1 and 2, and the address (group number) for standard (non-chained) multicast messages.

The values of the Ref1/2 Cycle Sel inputs correspond to the following intervals:

Value	Handling interval
0	Default (500 µs for reference 1; 2 ms for reference 2)
1	250 µs
2	500 µs
3	2 ms

Note: Negative value of Ref2 Cycle Sel disables the handling of Ref2 (if disabled in the master, it must be disabled in all follower drives as well).

Allowable values for the Std Mcast Group input are 0 (= multicasting not used) and 1…62 (multicast group).

An unconnected input, or an input in an error state, is interpreted as having the value 0.

The error codes indicated by the Error output are as follows:

Bit	Description
0	REF1_CYCLE_ERR: Value of input Ref1 Cycle Sel out of range
1	REF2_CYCLE_ERR: Value of input Ref2 Cycle Sel out of range
2	STD_MCAST_ERR: Value of input Std Mcast Group out of range

See also section *Examples of using standard function blocks in drive-to-drive communication* starting on page 339.

Inputs Drive-to-drive reference 1 handling interval (Ref1 Cycle Sel): INT

Drive-to-drive reference 2 handling interval (Ref2 Cycle Sel): INT

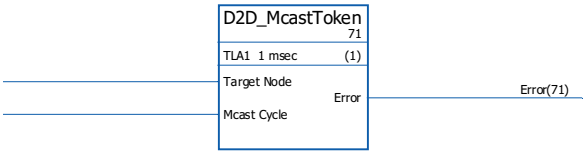
Standard multicast address (Std Mcast Group): INT

Outputs Error output (Error): PB

D2D_McastToken

(10096)

Illustration



Execution time -

Operation Configures the transmission of token messages sent to a follower. Each token authorizes the follower to send one message to another follower or group of followers. For the message types, see the block D2D_SendMessage.

Note: This block is only supported in the master.

The Target Node input defines the node address the master sends the tokens to; the range is 1…62.

The Mcast Cycle specifies the interval between token messages in the range of 2…1000 milliseconds. Setting this input to 0 disables the sending of tokens.

The error codes indicated by the Error output are as follows:

Bit	Description
0	D2D_MODE_ERR: Drive is not master
5	TOO_SHORT_CYCLE: Token interval is too short, causing overloading
6	INVALID_INPUT_VAL: An input value is out of range
7	GENERAL_D2D_ERR: Drive-to-drive communication driver failed to initialize message

See also section *Examples of using standard function blocks in drive-to-drive communication* starting on page 339.

Inputs Token recipient (Target Node): INT

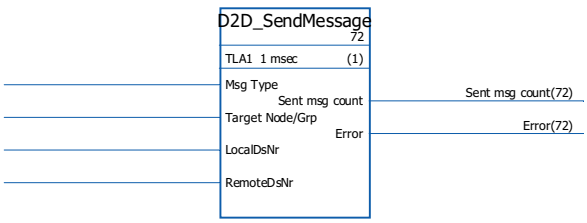
Token interval (Mcast Cycle): INT

Outputs Error output (Error): DINT

D2D_SendMessage

(10095)

Illustration



Execution time -

Operation

Configures the transmission between the dataset tables of drives.

The Msg Type input defines the message type as follows:

Value	Message type
0	Disabled
1	Master P2P: The master sends the contents of a local dataset (specified by LocalDsNr input) to the dataset table (dataset number specified by RemoteDsNr input) of a follower (specified by Target Node/Grp input). The follower replies by sending the next dataset (RemoteDsNr + 1) to the master (LocalDsNr + 1). The node number of a drive is defined by parameter 57.03. **Note:** Only supported in the master drive.
2	Read Remote: The master reads a dataset (specified by RemoteDsNr input) from a follower (specified by Target Node/Grp input) and stores it into local dataset table (dataset number specified by LocalDsNr input). The node number of a drive is defined by parameter 57.03. **Note:** Only supported in the master drive.
3	Follower P2P: The follower sends the contents of a local dataset (specified by LocalDsNr input) to the dataset table (dataset number specified by RemoteDsNr input) of another follower (specified by Target Node/Grp input). The node number of a drive is defined by parameter 57.03. **Note:** Only supported in a follower drive. A token from the master drive is required for the follower to be able to send the message. See the block D2D_McastToken.
4	Standard Multicast: The drive sends the contents of a local dataset (specified by LocalDsNr input) to the dataset table (dataset number specified by RemoteDsNr input) of a group of followers (specified by Target Node/Grp input). Which multicast group a drive belongs to is defined by the Std Mcast Group input of the D2D_Conf block. A token from the master drive is required for a follower to be able to send the message. See the block D2D_McastToken.
5	Broadcast: The drive sends the contents of a local dataset (specified by LocalDsNr input) to the dataset table (dataset number specified by RemoteDsNr input) of all followers. A token from the master drive is required for a follower to be able to send the message. See the block D2D_McastToken. **Note:** With this message type, the Target Node/Grp input must be connected in DriveSPC even if not used.

See also section *Examples of using standard function blocks in drive-to-drive communication* starting on page 339.

The Target Node/Grp input specifies the target drive or multicast group of drives depending on message type. See the message type explanations above.

Note: The input must be connected in DriveSPC even if not used.

The LocalDsNr input specifies the number of the local dataset used as the source or the target of the message.

The RemoteDsNr input specifies the number of the remote dataset used as the target or the source of the message.

The Sent msg count output is a wrap-around counter of successfully sent messages.

The error codes indicated by the Error output are as follows:

Bit	Description
0	D2D_MODE_ERR: Drive-to-drive communication not activated, or message type not supported in current drive-to-drive mode (master/follower)
1	LOCAL_DS_ERR: LocalDsNr input out of range (16…199)
2	TARGET_NODE_ERR: Target Node/Grp input out of range (1…62)
3	REMOTE_DS_ERR: Remote dataset number out of range (16…199)
4	MSG_TYPE_ERR: Msg Type input out of range (0…5)
5…6	Reserved
7	GENERAL_D2D_ERR: Unspecified error in D2D driver
8	RESPONSE_ERR: Syntax error in received response
9	TRA_PENDING: Message has not yet been sent
10	REC_PENDING: Response has not yet been received
11	REC_TIMEOUT: No response received
12	REC_ERROR: Frame error in received message
13	REJECTED: Message has been removed from transmit buffer
14	BUFFER_FULL: Transmit buffer full

Inputs

Message type (Msg Type): INT

Target node or multicast group (Target Node/Grp): INT

Local dataset number (LocalDsNr): INT

Remote dataset number (RemoteDsNr): INT

Outputs

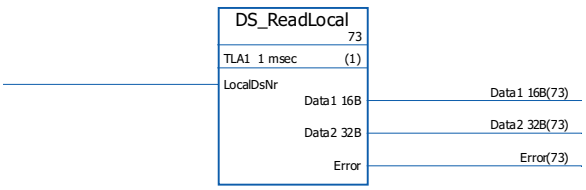
Successfully sent messages counter (Sent msg count): DINT

Error output (Error): PB

DS_ReadLocal

(10094)

Illustration



Execution time

-

Operation

Reads the dataset defined by the LocalDsNr input from the local dataset table. One dataset contains one 16-bit and one 32-bit word which are directed to the Data1 16B and Data2 32B outputs respectively.

The LocalDsNr input defines the number of the dataset to be read.

The error codes indicated by the Error output are as follows:

Bit	Description
1	LOCAL_DS_ERR: LocalDsNr out of range (16…199)

See also section *Examples of using standard function blocks in drive-to-drive communication* starting on page 339.

Inputs

Local dataset number (LocalDsNr): INT

Outputs

Contents of dataset (Data1 16B): INT

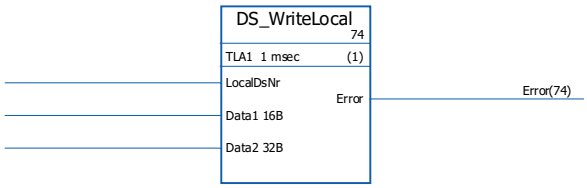
Contents of dataset (Data2 32B): DINT

Error output (Error): DINT

DS_WriteLocal

(10093)

Illustration



Execution time

-

Operation

Writes data into the local dataset table. Each dataset contains 48 bits; the data is input through the Data1 16B (16 bits) and Data2 32B (32 bits) inputs. The dataset number is defined by the LocalDsNr input.

The error codes indicated by the Error output are as follows:

Bit	Description
1	LOCAL_DS_ERR: LocalDsNr out of range (16…199)

See also section *Examples of using standard function blocks in drive-to-drive communication* starting on page 339.

Inputs

Local dataset number (LocalDsNr): INT

Contents of dataset (Data1 16B): INT

Contents of dataset (Data2 32B): DINT

Outputs

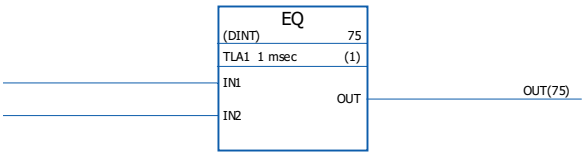
Error output (Error): DINT

Comparison

EQ

(10040)

Illustration



Execution time 0.89 µs (when two inputs are used) + 0.43 µs (for every additional input). When all inputs are used, the execution time is 13.87 µs.

Operation The output (OUT) is 1 if all the connected input values are equal (IN1 = IN2 = … = IN32). Otherwise the output is 0.

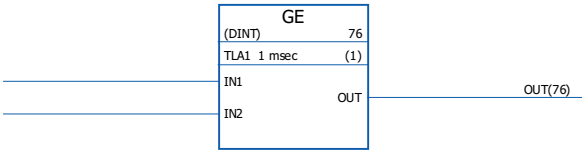
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): Boolean

GE >=

(10041)

Illustration



Execution time 0.89 µs (when two inputs are used) + 0.43 µs (for every additional input). When all inputs are used, the execution time is 13.87 µs.

Operation The output (OUT) is 1 if (IN1 ≥ IN2) & (IN2 ≥ IN3) & … & (IN31 ≥ IN32). Otherwise the output is 0.

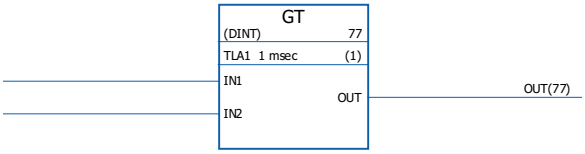
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): Boolean

GT >

(10042)

Illustration



Execution time 0.89 µs (when two inputs are used) + 0.43 µs (for every additional input). When all inputs are used, the execution time is 13.87 µs.

Operation The output (OUT) is 1 if (IN1 > IN2) & (IN2 > IN3) & … & (IN31 > IN32). Otherwise the output is 0.

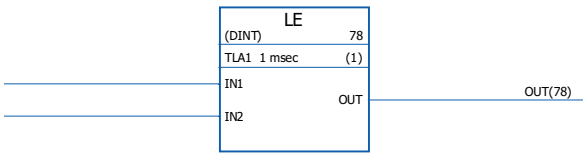
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): Boolean

LE (10043)

<=

Illustration



Execution time 0.89 µs (when two inputs are used) + 0.43 µs (for every additional input). When all inputs are used, the execution time is 13.87 µs.

Operation Output (OUT) is 1 if (IN1 ≤ IN2) & (IN2 ≤ IN3) & … & (IN31 ≤ IN32). Otherwise the output is 0.

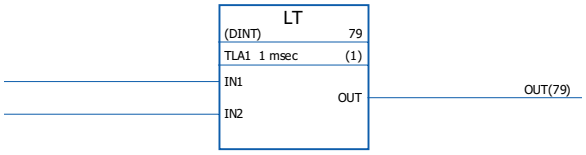
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): Boolean

LT (10044)

<

Illustration



Execution time 0.89 µs (when two inputs are used) + 0.43 µs (for every additional input). When all inputs are used, the execution time is 13.87 µs.

Operation Output (OUT) is 1 if (IN1 < IN2) & (IN2 < IN3) & … & (IN31 < IN32). Otherwise the output is 0.

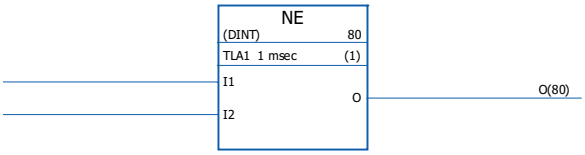
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): Boolean

NE <>

(10045)

Illustration



Execution time 0.44 µs

Operation The output (O) is 1 if I1 <> I2. Otherwise the output is 0.

Inputs The input data type is selected by the user.
Input (I1, I2): INT, DINT, REAL, REAL24

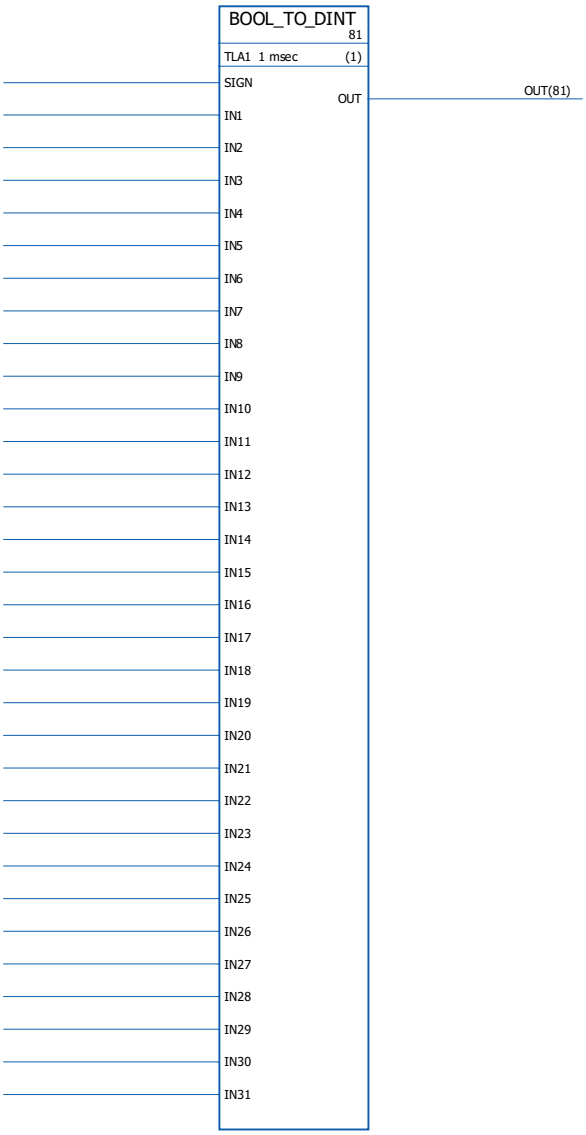
Outputs Output (O): Boolean

Conversion

BOOL_TO_DINT

(10018)

Illustration



Execution time 13.47 µs

Operation The output (OUT) value is a 32-bit integer value formed from the boolean input (IN1…IN31 and SIGN) values. IN1 = bit 0 and IN31 = bit 30.

Example:

IN1 = 1, IN2 = 0, IN3…IN31 = 1, SIGN = 1

OUT = 1111 1111 1111 1111 1111 1111 1111 1101

SIGN IN31…IN1

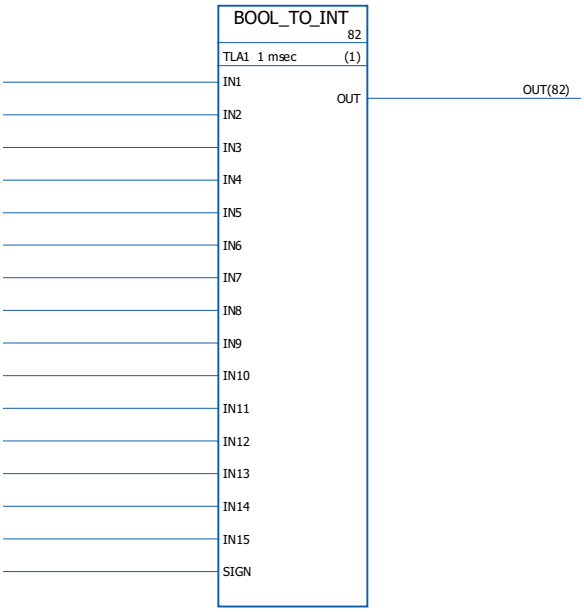
Input Sign input (SIGN): Boolean
Input (IN1…IN31): Boolean

Output Output (OUT): DINT (31 bits + sign)

BOOL_TO_INT

(10019)

Illustration



Execution time 5.00 µs

Operation The output (OUT) value is a 16-bit integer value formed from the boolean input (IN1…IN15 and SIGN) values. IN1 = bit 0 and IN15 = bit 14.

Example:

IN1…IN15 = 1, SIGN = 0

OUT = 0111 1111 1111 1111
SIGN IN15…IN1

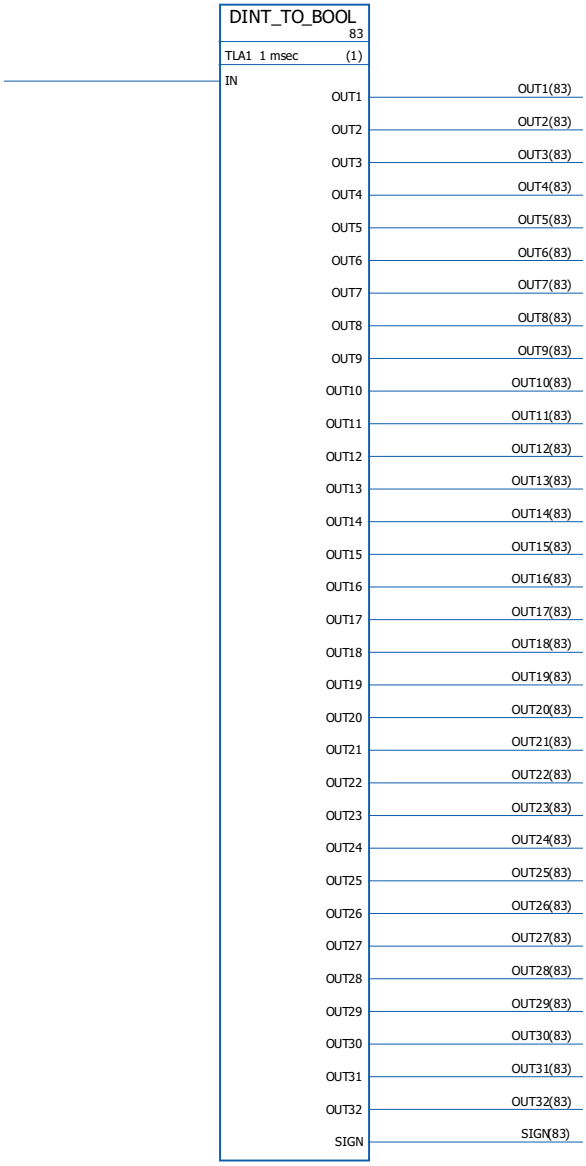
Inputs Input (IN1…IN15): Boolean
Sign input (SIGN): Boolean

Outputs Output (OUT): DINT (15 bits + sign)

DINT_TO_BOOL

(10020)

Illustration



Execution time 11.98 µs

Operation The boolean output (OUT1…OUT32) values are formed from the 32-bit integer input (IN) value.

Example:

IN = 0 111 1111 1111 1111 1111 1111 1111 1100

SIGN OUT32…OUT1

Inputs Input (IN): DINT

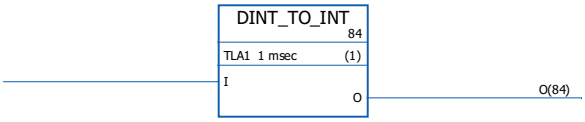
Outputs Output (OUT1…OUT32): Boolean

Sign output (SIGN): Boolean

DINT_TO_INT

(10021)

Illustration



Execution time 0.53 µs

Operation The output (O) value is a 16-bit integer value of the 32-bit integer input (I) value.

Examples:

I (31 bits + sign)	**O** (15 bits + sign)
2147483647	32767
-2147483648	-32767
0	0

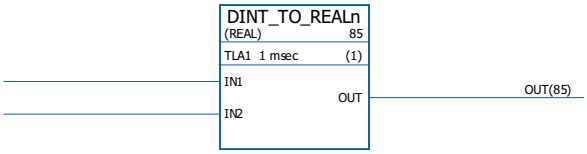
Inputs Input (I): DINT

Outputs Output (O): INT

DINT_TO_REALn

(10023)

Illustration



Execution time 7.25 µs

Operation The output (OUT) is the REAL/REAL24 equivalent of the input (IN). Input IN1 is the integer value and input IN2 is the fractional value.

If one (or both) of the input values is negative, the output value is negative.

Example (from DINT to REAL):

When IN1 = 2 and IN2 = 3276, OUT = 2.04999.

The output value is limited to the maximum value of the selected data type range.

Inputs Input (IN1, IN2): DINT

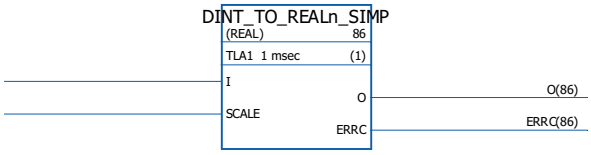
Outputs The output data type is selected by the user.

Output (OUT): REAL, REAL24

DINT_TO_REALn_SIMP

(10022)

Illustration



Execution time 6.53 µs

Operation The output (O) is the REAL/REAL24 equivalent of the input (I) divided by the scale input (SCALE).

Error codes indicated at the error output (ERRC) are as follows:

Error code	Description
0	No error
1001	The calculated REAL/REAL24 value exceeds the minimum value of the selected data type range. The output is set to the minimum value.
1002	The calculated REAL/REAL24 value exceeds the maximum value of the selected data type range. The output is set to the maximum value.
1003	The SCALE input is 0. The output is set to 0.
1004	Incorrect SCALE input, i.e. the scale input is < 0 or is not a factor of 10.

Example (from DINT to REAL24):

When I = 205 and SCALE = 100, I/SCALE = 205 /100 = 2.05 and O = 2.04999.

Inputs Input (I): DINT

Scale input (SCALE): DINT

Outputs The output data type is selected by the user.

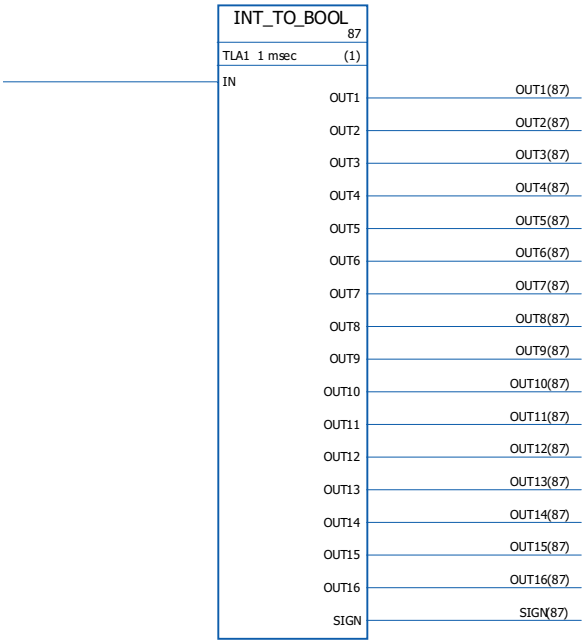
Output (O): REAL, REAL24

Error output (ERRC): DINT

INT_TO_BOOL

(10024)

Illustration



Execution time 4.31 µs

Operation The boolean output (OUT1…OUT16) values are formed from the 16-bit integer input (IN) value.

Example:

IN = 0111 1111 1111 1111

SIGN OUT16…OUT1

Inputs Input (IN): INT

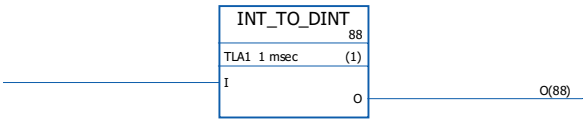
Outputs Output (OUT1…OUT16): Boolean

Sign output (SIGN): Boolean

INT_TO_DINT

(10025)

Illustration



Execution time 0.33 µs

Operation The output (O) value is a 32-bit integer value of the 16-bit integer input (I) value.

I	O
32767	32767
-32767	-32767
0	0

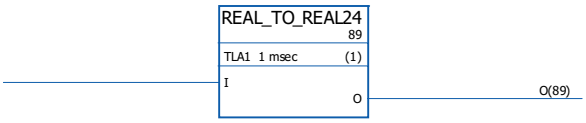
Inputs Input (I): INT

Outputs Output (O): DINT

REAL_TO_REAL24

(10026)

Illustration



Execution time 1.35 µs

Operation Output (O) is the REAL24 equivalent of the REAL input (I).
The output value is limited to the maximum value of the data type.
Example:

I = 0000 0000 0010 0110 1111 1111 1111 1111
(Integer value / Fractional value)

O = 0010 0110 1111 1111 1111 1111 0000 0000
(Integer value / Fractional value)

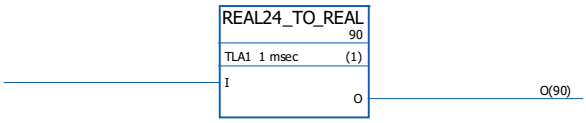
Inputs Input (I): REAL

Outputs Output (O): REAL24

REAL24_TO_REAL

(10027)

Illustration



Execution time 1.20 µs

Operation Output (O) is the REAL equivalent of the REAL24 input (I).
The output value is limited to the maximum value of the data type range.
Example:

I = 0010 0110 1111 1111 1111 1111 0000 0000
(Integer value / Fractional value)

O = 0000 0000 0010 0110 1111 1111 1111 1111
(Integer value / Fractional value)

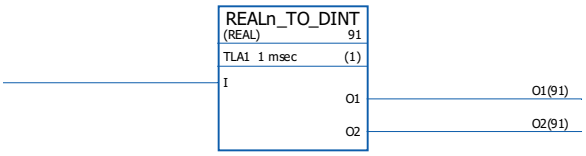
Inputs Input (I): REAL24

Outputs Output (O): REAL

REALn_TO_DINT

(10029)

Illustration



Execution time 6.45 µs

Operation Output (O) is the 32-bit integer equivalent of the REAL/REAL24 input (I). Output O1 is the integer value and output O2 is the fractional value.

The output value is limited to the maximum value of the data type range.

Example (from REAL to DINT):

When I = 2.04998779297, O1 = 2 and O2 = 3276.

Inputs The input data type is selected by the user.

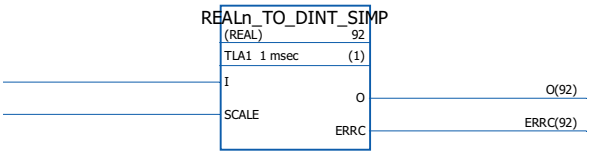
Input (I): REAL, REAL24

Outputs Output (O1, O2): DINT

REALn_TO_DINT_SIMP

(10028)

Illustration



Execution time 5.54 µs

Operation Output (O) is the 32-bit integer equivalent of the REAL/REAL24 input (I) multiplied by the scale input (SCALE).

Error codes are indicated by the error output (ERRC) as follows:

Error code	Description
0	No error
1001	The calculated integer value exceeds the minimum value. The output is set to the minimum value.
1002	The calculated integer value exceeds the maximum value. The output is set to the maximum value.
1003	Scale input is 0. The output is set to 0.
1004	Incorrect scale input, i.e. scale input is < 0 or is not a factor of 10.

Example (from REAL to DINT):

When I = 2.04998779297and SCALE = 100, O = 204.

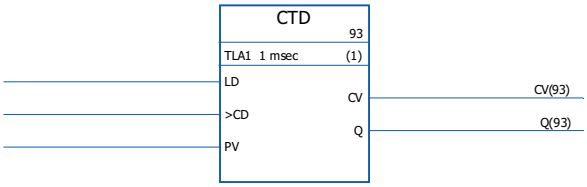
Inputs	The input data type is selected by the user. Input (I): REAL, REAL24 Scale input (SCALE): DINT
Outputs	Output (O): DINT Error output (ERRC): DINT

Counters

CTD

(10047)

Illustration



Execution time 0.92 µs

Operation The counter output (CV) value is decreased by 1 if the counter input (CD) value changes from 0 -> 1 and the load input (LD) value is 0. If the load input value is 1, the preset input (PV) value is stored as the counter output (CV) value. If the counter output has reached its minimum value -32768, the counter output remains unchanged.

The status output (Q) is 1 if the counter output (CV) value ≤ 0.

Example:

LD	CD	PV	Q	CV_{prev}	CV
0	1 -> 0	10	0	5	5
0	0 -> 1	10	0	5	5 - 1 = 4
1	1 -> 0	-2	1	4	-2
1	0 -> 1	1	0	-2	1
0	0 -> 1	5	1	1	1 -1 = 0
1	1 -> 0	-32768	1	0	-32768
0	0 -> 1	10	1	-32768	-32768
CV_{prev} is the previous cycle counter output value.					

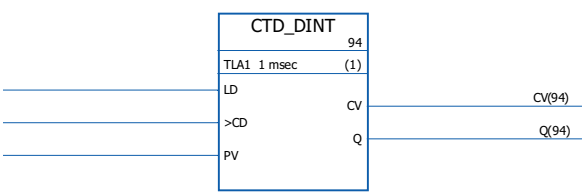
Inputs Load input (LD): Boolean
Counter input (CD): Boolean
Preset input (PV): INT

Outputs Counter output (CV): INT
Status output (Q): Boolean

CTD_DINT

(10046)

Illustration



Execution time 0.92 µs

Operation

The counter output (CV) value is decreased by 1 if the counter input (CD) value changes from 0 -> 1 and the load input (LD) value is 0. If the load input (LD) value is 1, the preset input (PV) value is stored as the counter output (CV) value. If the counter output has reached its minimum value -2147483648, the counter output remains unchanged.

The status output (Q) is 1 if the counter output (CV) value ≤ 0.

Example:

LD	**CD**	**PV**	**Q**	**CV_{prev}**	**CV**
0	1 -> 0	10	0	5	5
0	0 -> 1	10	0	5	5 - 1 = 4
1	1 -> 0	-2	1	4	-2
1	0 -> 1	1	0	-2	1
0	0 -> 1	5	1	1	1 -1 = 0
1	1 -> 0	-2147483648	1	0	-2147483648
0	0 -> 1	10	1	-2147483648	-2147483648
CV_{prev} is the previous cycle counter output value.					

Inputs

Load input (LD): Boolean

Counter input (CD): Boolean

Preset input (PV): DINT

Outputs

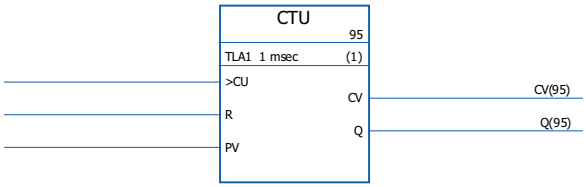
Counter output (CV): DINT

Status output (Q): Boolean

CTU

(10049)

Illustration



Execution time

0.92 µs

Operation

The counter output (CV) value is increased by 1 if the counter input (CU) value changes from 0 -> 1 and the reset input (R) value is 0. If the counter output has reached its maximum value 32767, the counter output remains unchanged.

The counter output (CV) is reset to 0 if the reset input (R) is 1.

The status output (Q) is 1 if the counter output (CV) value ≥ preset input (PV) value.

Example:

R	CU	PV	Q	CV_{prev}	CV
0	1 -> 0	20	0	10	10
0	0 -> 1	11	1	10	10 + 1 = 11
1	1 -> 0	20	0	11	0
1	0 -> 1	5	0	0	0
0	0 -> 1	20	0	0	0 + 1 = 1
0	0 -> 1	30	1	32767	32767
CV_{prev} is the previous cycle counter output value.					

Inputs

Counter input (CU): Boolean

Reset input (R): Boolean

Preset input (PV): INT

Outputs

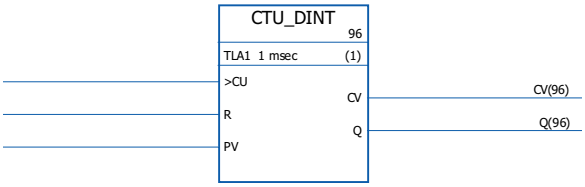
Counter output (CV): INT

Status output (Q): Boolean

CTU_DINT

(10048)

Illustration



Execution time

0.92 µs

Operation

The counter output (CV) value is increased by 1 if the counter input (CU) value changes from 0 -> 1 and the reset input (R) value is 0. If the counter output has reached its maximum value 2147483647, the counter output remains unchanged.

The counter output (CV) is reset to 0 if the reset input (R) is 1.

The status output (Q) is 1 if the counter output (CV) value ≥ preset input (PV) value.

Example:

R	CU	PV	Q	CV_{prev}	CV
0	1 -> 0	20	0	10	10
0	0 -> 1	11	1	10	10 + 1 = 11
1	1 -> 0	20	0	11	0
1	0 -> 1	5	0	0	0
0	0 -> 1	20	0	0	0 + 1 = 1
0	0 -> 1	30	1	2147483647	2147483647
CV_{prev} is the previous cycle counter output value.					

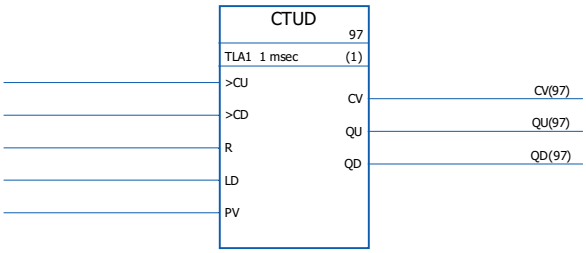
Inputs Counter input (CU): Boolean
Reset input (R): Boolean
Preset input (PV): DINT

Outputs Counter output (CV): DINT
Status output (Q): Boolean

CTUD

(10051)

Illustration



Execution time 1.40 µs

Operation

The counter output (CV) value is increased by 1 if the counter input (CU) value changes from 0 -> 1 and the reset input (R) is 0 and the load input (LD) is 0.

The counter output (CV) value is decreased by 1 if the counter input (CD) changes from 0 -> 1 and the load input (LD) is 0 and the reset input (R) is 0.

If the load input (LD) is 1, the preset input (PV) value is stored as the counter output (CV) value.

The counter output (CV) is reset to 0 if the reset input (R) is 1.

If the counter output has reached its minimum or maximum value, -32768 or +32767, the counter output remains unchanged until it is reset (R) or until the load input (LD) is set to 1.

The up counter status output (QU) is 1 if the counter output (CV) value $\geq$ preset input (PV) value.

The down counter status output (QD) is 1 if the counter output (CV) value $\leq$ 0.

Example:

CU	**CD**	**R**	**LD**	**PV**	**QU**	**QD**	**CV_{prev}**	**CV**
0 -> 0	0 -> 0	0	0	2	0	1	0	0
0 -> 0	0 -> 0	0	1	2	1	0	0	2
0 -> 0	0 -> 0	1	0	2	0	1	2	0
0 -> 0	0 -> 0	1	1	2	0	1	0	0
0 -> 0	0 -> 1	0	0	2	0	1	0	0 - 1 = -1
0 -> 0	1 -> 1	0	1	2	1	0	-1	2
0 -> 0	1 -> 1	1	0	2	0	1	2	0
0 -> 0	1 -> 1	1	1	2	0	1	0	0
0 -> 1	1 -> 0	0	0	2	0	0	0	0 + 1 = 1
1 -> 1	0 -> 0	0	1	2	1	0	1	2
1 -> 1	0 -> 0	1	0	2	0	1	2	0
1 -> 1	0 -> 0	1	1	2	0	1	0	0
1 -> 1	0 -> 1	0	0	2	0	1	0	0 - 1 = -1
1 -> 1	1 -> 1	0	1	2	1	0	-1	2
1 -> 1	1 -> 1	1	0	2	0	1	2	0
1 -> 1	1 -> 1	1	1	2	0	1	0	0
CV_{prev} is the previous cycle counter output value.								

Inputs

Up counter input (CU): Boolean

Down counter input (CD): Boolean

Reset input (R): Boolean

Load input (LD): Boolean

Preset input (PV): INT

Outputs

Counter output (CV): INT

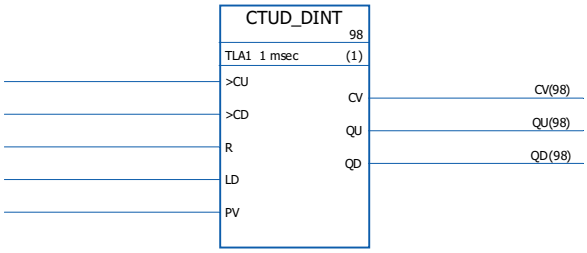
Up counter status output (QU): Boolean

Down counter status output (QD): Boolean

CTUD_DINT

(10050)

Illustration



Execution time 1.40 µs

Operation

The counter output (CV) value is increased by 1 if the counter input (CU) changes from 0 -> 1 and the reset input (R) is 0 and the load input (LD) is 0.

The counter output (CV) value is decreased by 1 if the counter input (CD) changes from 0 -> 1 and the load input (LD) is 0 and the reset input (R) is 0.

If the counter output has reached its minimum or maximum value, -2147483648 or +2147483647, the counter output remains unchanged until it is reset (R) or until the load input (LD) is set to 1.

If the load input (LD) value is 1, the preset input (PV) value is stored as the counter output (CV) value.

The counter output (CV) is reset to 0 if the reset input (R) is 1.

The up counter status output (QU) is 1 if the counter output (CV) value $\geq$ preset input (PV) value.

The down counter status output (QD) is 1 if the counter output (CV) value $\leq$ 0.

Example:

CU	CD	R	LD	PV	QU	QD	CV_{prev}	CV
0 -> 0	0 -> 0	0	0	2	0	1	0	0
0 -> 0	0 -> 0	0	1	2	1	0	0	2
0 -> 0	0 -> 0	1	0	2	0	1	2	0
0 -> 0	0 -> 0	1	1	2	0	1	0	0
0 -> 0	0 -> 1	0	0	2	0	1	0	0 - 1 = -1
0 -> 0	1 -> 1	0	1	2	1	0	-1	2
0 -> 0	1 -> 1	1	0	2	0	1	2	0
0 -> 0	1 -> 1	1	1	2	0	1	0	0
0 -> 1	1 -> 0	0	0	2	0	0	0	0 + 1 = 1
1 -> 1	0 -> 0	0	1	2	1	0	1	2
1 -> 1	0 -> 0	1	0	2	0	1	2	0
1 -> 1	0 -> 0	1	1	2	0	1	0	0
1 -> 1	0 -> 1	0	0	2	0	1	0	0 - 1 = -1
1 -> 1	1 -> 1	0	1	2	1	0	-1	2
1 -> 1	1 -> 1	1	0	2	0	1	2	0
1 -> 1	1 -> 1	1	1	2	0	1	0	0
CV_{prev} is the previous cycle counter output value.								

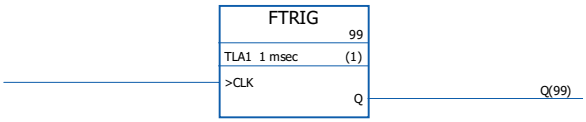
Inputs	Up counter input (CU): Boolean Down counter input (CD): Boolean Reset input (R): Boolean Load input (LD): Boolean Preset input (PV): DINT
Outputs	Counter output (CV): DINT Up counter status output (QU): Boolean Down counter status output (QD): Boolean

Edge & bistable

FTRIG

(10030)

Illustration



Execution time 0.38 µs

Operation The output (Q) is set to 1 when the clock input (CLK) changes from 1 to 0. The output is set back to 0 with the next execution of the block. Otherwise the output is 0.

$CLK_{previous}$	**CLK**	**Q**
0	0	0
0	1	0
1	0	1 (for one execution cycle time, returns to 0 at the next execution)
1	1	0
$CLK_{previous}$ is the previous cycle output value.		

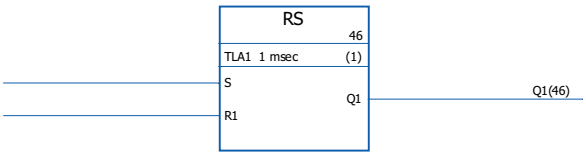
Inputs Clock input (CLK): Boolean

Outputs Output (Q): Boolean

RS

(10032)

Illustration



Execution time 0.38 µs

Operation The output (Q1) is 1 if the set input (S) is 1 and the reset input (R1) is 0. The output will retain the previous output state if the set input (S) and the reset input (R1) are 0. The output is 0 if the reset input is 1.

Truth table:

S	R1	$Q1_{previous}$	Q1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0
$Q_{previous}$ is the previous cycle output value.			

Inputs Set input (S): Boolean

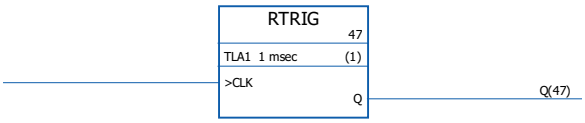
Reset input (R1): Boolean

Outputs Output (Q1): Boolean

RTRIG

(10031)

Illustration



Execution time 0.38 µs

Operation The output (Q) is set to 1 when the clock input (CLK) changes from 0 to 1. The output is set back to 0 with the next execution of the block. Otherwise the output is 0.

$CLK_{previous}$	CLK	Q
0	0	0
0	1	1
1	0	0
1	1	0
$CLK_{previous}$ is the previous cycle output value.		

Note: The output (Q) is 1 after the first execution of the block after cold restart when the clock input (CLK) is 1. Otherwise the output is always 0 when the clock input is 1.

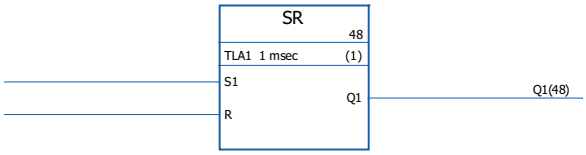
Inputs Clock input (CLK): Boolean

Outputs Output (Q): Boolean

SR

(10033)

Illustration



Execution time 0.38 µs

Operation The output (Q1) is 1 if the set input (S1) is 1. The output will retain the previous output state if the set input (S1) and the reset input (R) are 0. The output is 0 if the set input is 0 and the reset input is 1.

Truth table:

S1	**R**	**$Q1_{previous}$**	**Q1**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1
$Q1_{previous}$ is the previous cycle output value.			

Inputs Set input (S1): Boolean

Reset input (R): Boolean

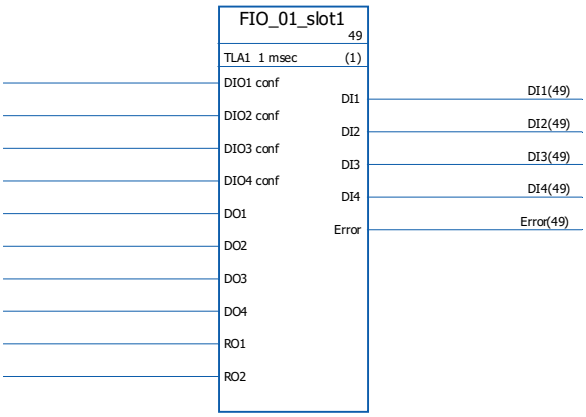
Outputs Output (Q1): Boolean

Extensions

FIO_01_slot1

(10084)

Illustration



Execution time 8.6 µs

Operation The block controls the four digital inputs/outputs (DIO1…DIO4) and two relay outputs (RO1, RO2) of a FIO-01 Digital I/O Extension mounted on slot 1 of the drive control unit.

The state of a DIOx conf input of the block determines whether the corresponding DIO on the FIO-01 is an input or an output (0 = input, 1 = output). If the DIO is an output, the DOx input of the block defines its state.

The RO1 and RO2 inputs define the state of the relay outputs of the FIO-01 (0 = not energised, 1 = energised).

The DIx outputs show the state of the DIOs.

Inputs Digital input/output mode selection (DIO1 conf … DIO4 conf): Boolean

Digital output state selection (DO1…DO4): Boolean

Relay output state selection (RO1, RO2): Boolean

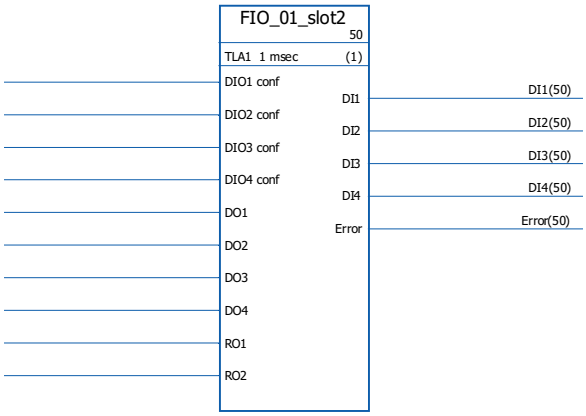
Outputs Digital input/output state (DI1…DI4): Boolean

Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_01_slot2

(10085)

Illustration



Execution time 8.6 µs

Operation The block controls the four digital inputs/outputs (DIO1…DIO4) and two relay outputs (RO1, RO2) of a FIO-01 Digital I/O Extension mounted on slot 2 of the drive control unit.

The state of a DIOx conf input of the block determines whether the corresponding DIO on the FIO-01 is an input or an output (0 = input, 1 = output). If the DIO is an output, the DOx input of the block defines its state.

The RO1 and RO2 inputs define the state of the relay outputs of the FIO-01 (0 = not energised, 1 = energised).

The DIx outputs show the state of the DIOs.

Inputs Digital input/output mode selection (DIO1 conf … DIO4 conf): Boolean

Digital output state selection (DO1…DO4): Boolean

Relay output state selection (RO1, RO2): Boolean

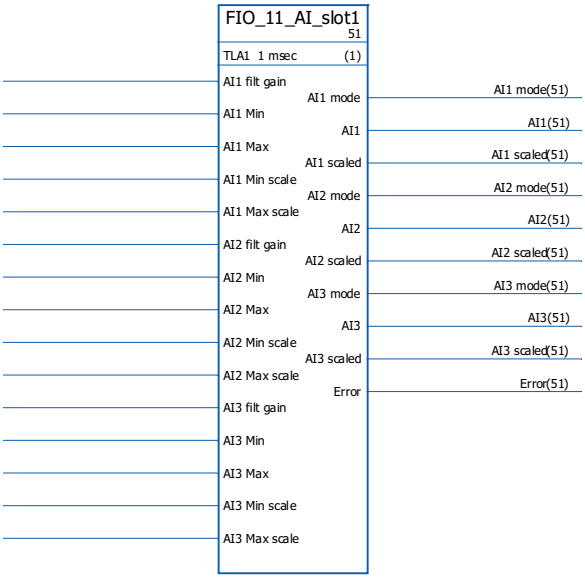
Outputs Digital input/output state (DI1…DI4): Boolean

Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_11_AI_slot1

(10088)

Illustration



Execution time 11.1 µs

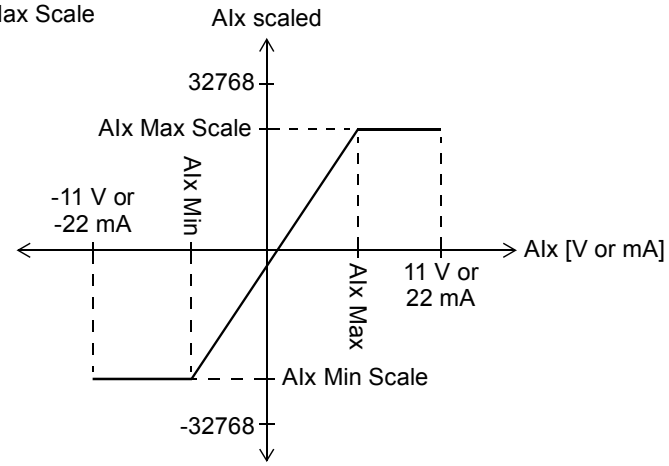
Operation

The block controls the three analogue inputs (AI1…AI3) of a FIO-11 Analog I/O Extension mounted on slot 1 of the drive control unit.

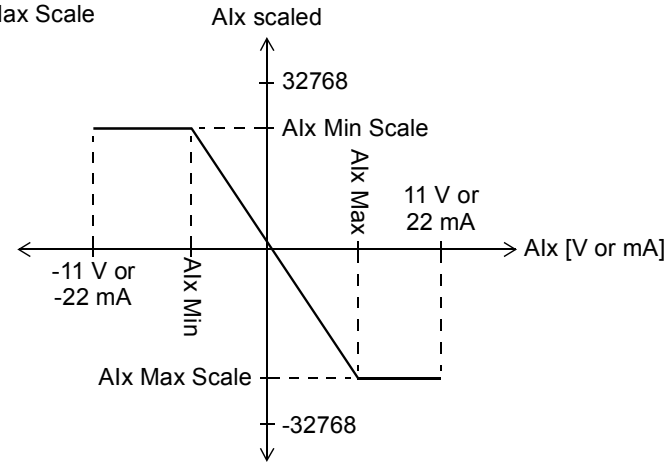
The block outputs both the unscaled (AIx) and scaled (AIx scaled) actual values of each analogue input. The scaling is based on the relationship between the ranges AIx min … AIx max and AIx min scale … AIx max scale.

AIx Min must be smaller than AIx Max; AIx Max Scale can be greater or smaller than AIx Min Scale.

AIx Min Scale < AIx Max Scale



AIx Min Scale > AIx Max Scale



The AIx filt gain inputs determine a filtering time for each input as follows:

AIx filt gain	Filtering time	Notes
0	No filtering	
1	125 µs	Recommended setting
2	250 µs	
3	500 µs	
4	1 ms	
5	2 ms	
6	4 ms	
7	7.9375 ms	

The AIx mode outputs show whether the corresponding input is voltage (0) or current (1). The voltage/current selection is made using the hardware switches on the FIO-11.

Inputs

Analogue input filter gain selection (AI1 filt gain … AI3 filt gain): INT
Minimum value of input signal (AI1 Min … AI3 Min): REAL (≥ -11 V or -22 mA)
Maximum value of input signal (AI1 Max … AI3 Max): REAL (≤ 11 V or 22 mA)
Minimum value of scaled output signal (AI1 Min scale … AI3 Min scale): REAL
Maximum value of scaled output signal (AI1 Max scale … AI3 Max scale): REAL

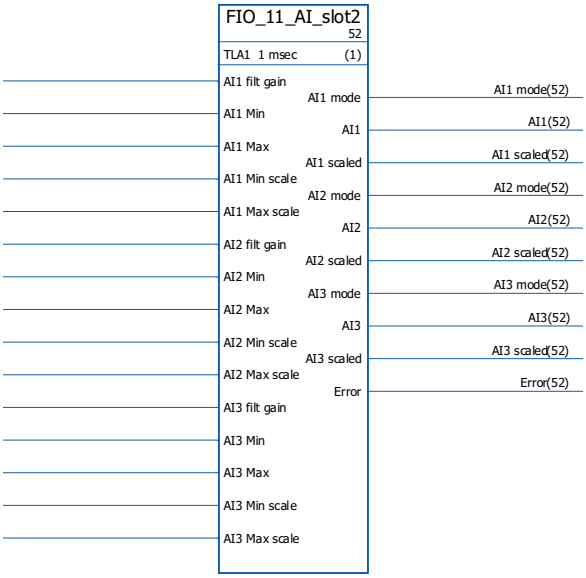
Outputs

Analogue input mode (voltage or current) (AI1 mode … AI3 mode): Boolean
Value of analogue input (AI1 … AI3): REAL
Scaled value of analogue input (AI1 scaled … AI3 scaled): REAL
Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_11_AI_slot2

(10089)

Illustration



Execution time 11.1 µs

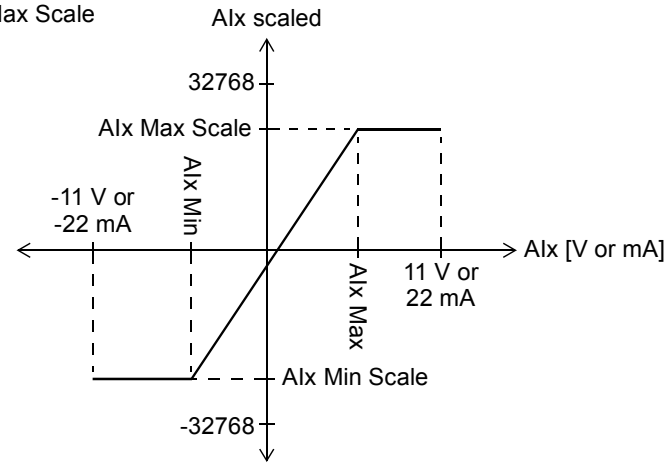
Operation

The block controls the three analogue inputs (AI1…AI3) of a FIO-11 Analog I/O Extension mounted on slot 2 of the drive control unit.

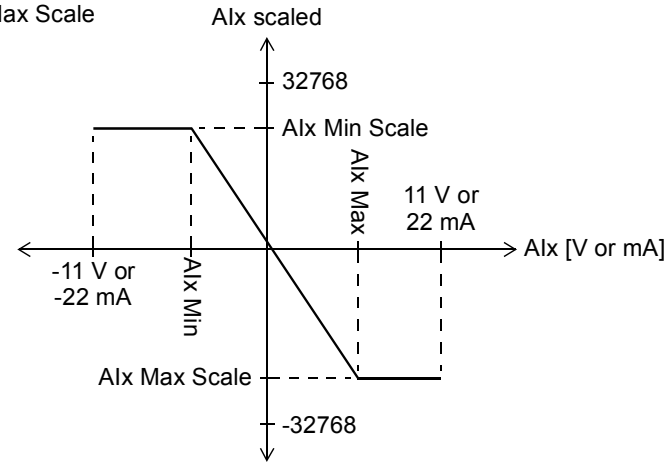
The block outputs both the unscaled (AIx) and scaled (AIx scaled) actual values of each analogue input. The scaling is based on the relationship between the ranges AIx min … AIx max and AIx min scale … AIx max scale.

AIx Min must be smaller than AIx Max; AIx Max Scale can be greater or smaller than AIx Min Scale.

AIx Min Scale < AIx Max Scale



AIx Min Scale > AIx Max Scale



The AIx filt gain inputs determine a filtering time for each input as follows:

AIx filt gain	Filtering time	Notes
0	No filtering	
1	125 µs	Recommended setting
2	250 µs	
3	500 µs	
4	1 ms	
5	2 ms	
6	4 ms	
7	7.9375 ms	

The AIx mode outputs show whether the corresponding input is voltage (0) or current (1). The voltage/current selection is made using the hardware switches on the FIO-11.

Inputs

Analogue input filter gain selection (AI1 filt gain … AI3 filt gain): INT
Minimum value of input signal (AI1 Min … AI3 Min): REAL (≥ -11 V or -22 mA)
Maximum value of input signal (AI1 Max … AI3 Max): REAL (≤ 11 V or 22 mA)
Minimum value of scaled output signal (AI1 Min scale … AI3 Min scale): REAL
Maximum value of scaled output signal (AI1 Max scale … AI3 Max scale): REAL

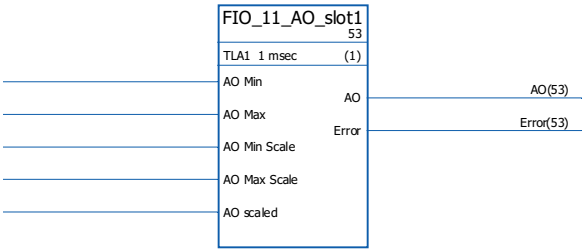
Outputs

Analogue input mode (voltage or current) (AI1 mode … AI3 mode): Boolean
Value of analogue input (AI1 … AI3): REAL
Scaled value of analogue input (AI1 scaled … AI3 scaled): REAL
Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_11_AO_slot1

(10090)

Illustration



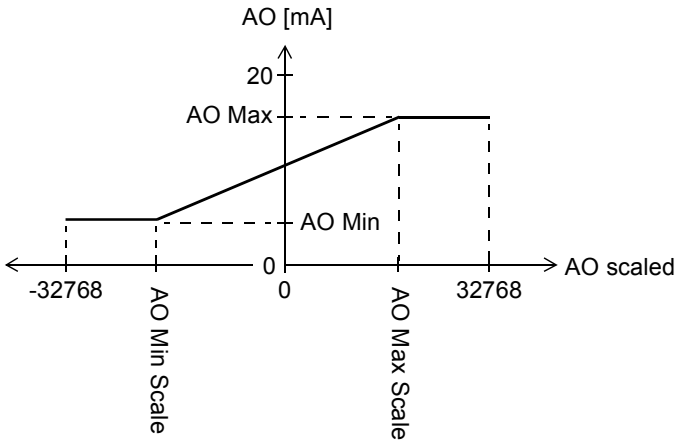
Execution time 4.9 µs

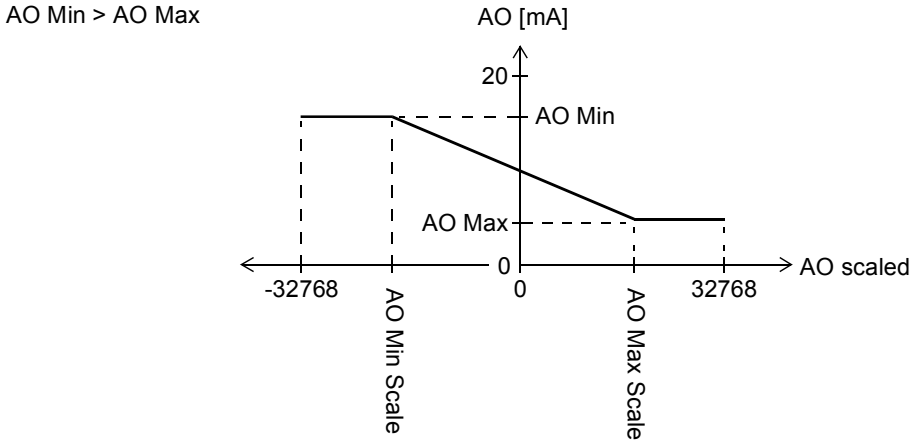
Operation The block controls the analogue output (AO1) of a FIO-11 Analog I/O Extension mounted on slot 1 of the drive control unit.

The block converts the input signal (AO scaled) to a 0…20 mA signal (AO) that drives the analogue output; the input range AO Min Scale … AO Max Scale corresponds to the current signal range of AO Min … AO Max.

AO Min Scale must be smaller than AO Max Scale; AO Max can be greater or smaller than AO Min.

AO Min < AO Max





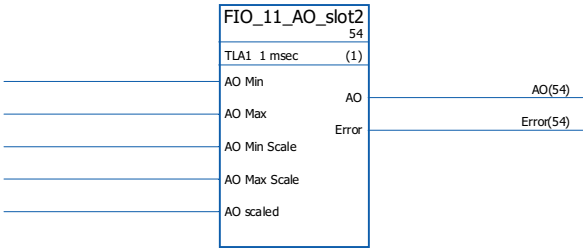
Inputs
Minimum current signal (AO Min): REAL (0…20 mA)
Maximum current signal (AO Max): REAL (0…20 mA)
Minimum input signal (AO Min Scale): REAL
Maximum input signal (AO Max Scale): REAL
Input signal (AO scaled): REAL

Outputs
Analogue output current value (AO): REAL
Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_11_AO_slot2

(10091)

Illustration



Execution time 4.9 µs

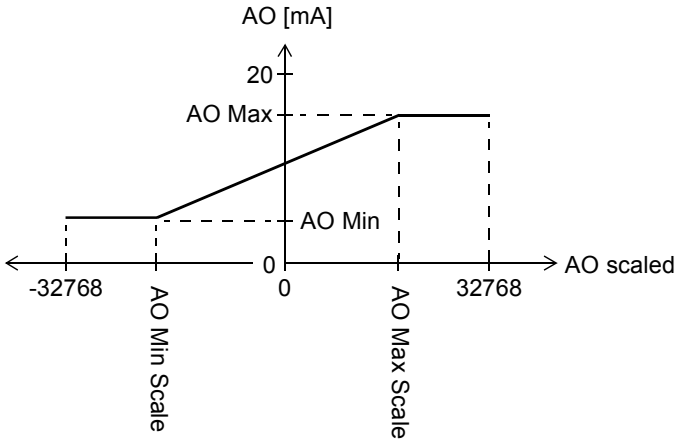
Operation

The block controls the analogue output (AO1) of a FIO-11 Analog I/O Extension mounted on slot 2 of the drive control unit.

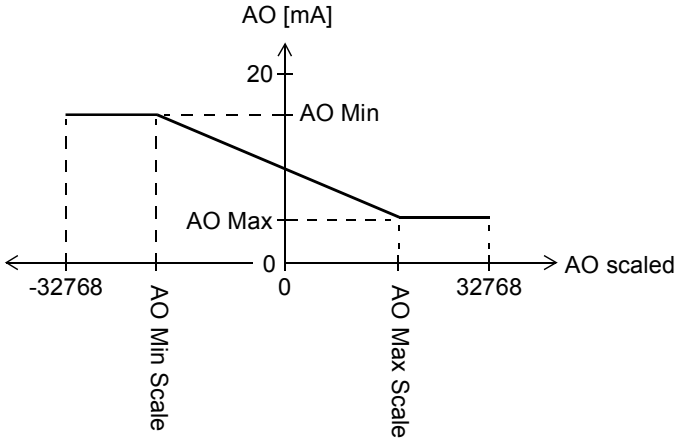
The block converts the input signal (AO scaled) to a 0…20 mA signal (AO) that drives the analogue output; the input range AO Min Scale … AO Max Scale corresponds to the current signal range of AO Min … AO Max.

AO Min Scale must be smaller than AO Max Scale; AO Max can be greater or smaller than AO Min.

AO Min < AO Max



AO Min > AO Max



Inputs

Minimum current signal (AO Min): REAL (0…20 mA)

Maximum current signal (AO Max): REAL (0…20 mA)

Minimum input signal (AO Min Scale): REAL

Maximum input signal (AO Max Scale): REAL

Input signal (AO scaled): REAL

Outputs

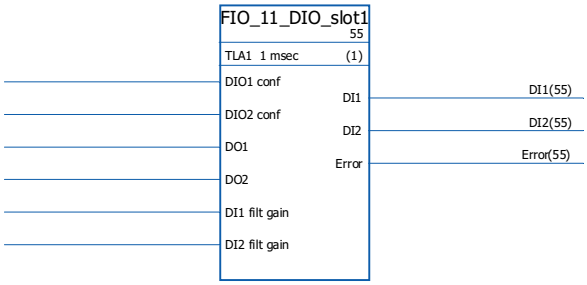
Analogue output current value (AO): REAL

Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_11_DIO_slot1

(10086)

Illustration



Execution time 6.0 µs

Operation The block controls the two digital inputs/outputs (DIO1, DIO2) of a FIO-11 Digital I/O Extension mounted on slot 1 of the drive control unit.

The state of a DIOx conf input of the block determines whether the corresponding DIO on the FIO-11 is an input or an output (0 = input, 1 = output). If the DIO is an output, the DOx input of the block defines its state.

The DIx outputs show the state of the DIOs.

The DIx filt gain inputs determine a filtering time for each input as follows:

DIx filt gain	Filtering time
0	7.5 µs
1	195 µs
2	780 µs
3	4.680 ms

Inputs Digital input/output mode selection (DIO1 conf, DIO2 conf): Boolean

Digital output state selection (DO1, DO2): Boolean

Digital input filter gain selection (DI1 filt gain, DI2 filt gain): INT

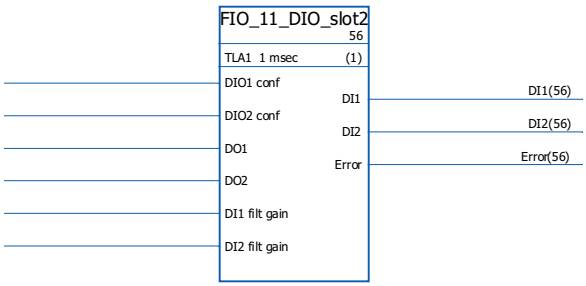
Outputs Digital input/output state (DI1, DI2): Boolean

Error output (Error): DINT (0 = No error; 1 = Application program memory full)

FIO_11_DIO_slot2

(10087)

Illustration



Execution time 6.0 µs

Operation

The block controls the two digital inputs/outputs (DIO1, DIO2) of a FIO-11 Digital I/O Extension mounted on slot 2 of the drive control unit.

The state of a DIOx conf input of the block determines whether the corresponding DIO on the FIO-11 is an input or an output (0 = input, 1 = output). If the DIO is an output, the DOx input of the block defines its state.

The DIx outputs show the state of the DIOs.

The DIx filt gain inputs determine a filtering time for each input as follows:

DIx filt gain	Filtering time
0	7.5 µs
1	195 µs
2	780 µs
3	4.680 ms

Inputs

Digital input/output mode selection (DIO1 conf, DIO2 conf): Boolean

Digital output state selection (DO1, DO2): Boolean

Digital input filter gain selection (DI1 filt gain, DI2 filt gain): INT

Outputs

Digital input/output state (DI1, DI2): Boolean

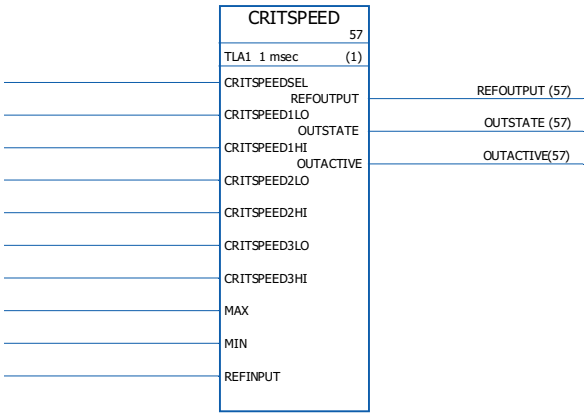
Error output (Error): DINT (0 = No error; 1 = Application program memory full)

Feedback & algorithms

CRITSPEED

(10068)

Illustration



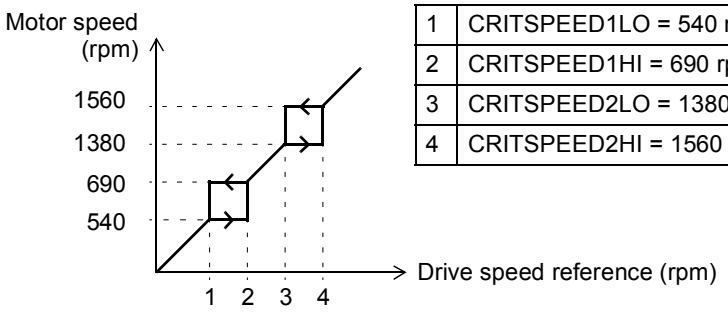
Execution time 4.50 µs

Operation

A critical speeds function block is available for applications where it is necessary to avoid certain motor speeds or speed bands because of e.g. mechanical resonance problems. The user can define three critical speeds or speed bands.

Example: An application has vibrations in the range of 540 to 690 rpm and 1380 to 1560 rpm. To make the drive made to jump over the vibration speed ranges:

- activate the critical speeds function (CRITSPEEDSEL = 1),
- set the critical speed ranges as in the figure below.



1	CRITSPEED1LO = 540 rpm
2	CRITSPEED1HI = 690 rpm
3	CRITSPEED2LO = 1380 rpm
4	CRITSPEED2HI = 1560 rpm

Output OUTACTIVE is set to 1 when the output reference (REFOUTPUT) is different from the input reference (REFINPUT).

The output is limited by the defined minimum and maximum limits (MIN and MAX).

Output OUTSTATE indicates in which critical speed range the operation point is.

Inputs

Critical speed activation input (CRITSPEEDSEL): Boolean

Minimum/maximum critical speed range input (CRITSPEEDNLO / CRITSPEEDNHI): REAL

Maximum/minimum input (MAX/MIN): REAL

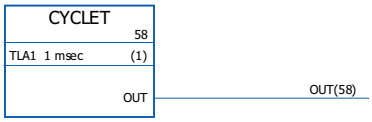
Reference input (REFINPUT): REAL

Outputs Reference output (REFOUTPUT): REAL
Output state (OUTSTATE): REAL
Output active (OUTACTIVE): Boolean

CYCLET

(10074)

Illustration



Execution time 0.00 µs

Operation Output (OUT) is the time level of the CYCLET function block.

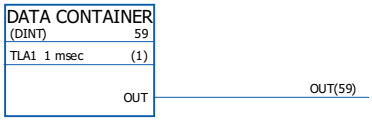
Inputs -

Outputs Output (OUT): DINT. 1 = 1 µs

DATA CONTAINER

(10073)

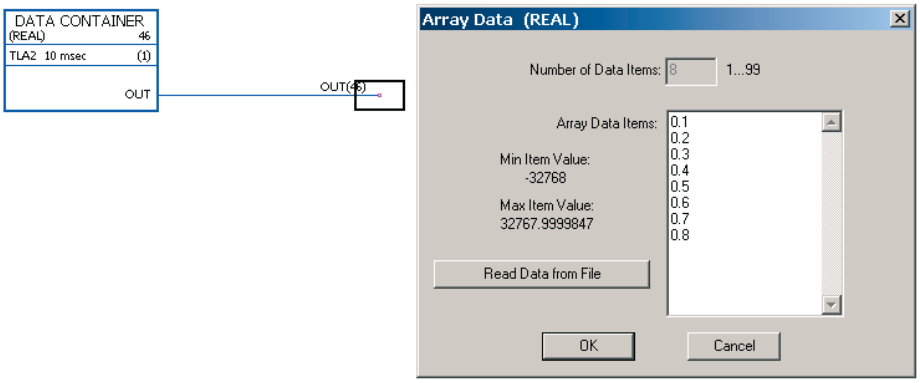
Illustration



Execution time 0.00 µs

Operation Output (OUT) is an array of data with values 1…99. The array can be used by the XTAB and YTAB tables in the block FUNG-1V (page 286). The array is defined by selecting “Define Pin Array Data” on the output pin in DriveSPC. Each value in the array must be on a separate row. Data can also be read from an *.arr file.

Example:



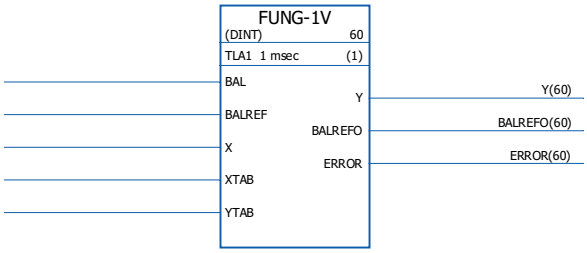
Inputs -

Outputs The output data type and the number of coordinate pairs are selected by the user.
Output (OUT): DINT, INT, REAL or REAL24

FUNG-1V

(10072)

Illustration



Execution time 9.29 µs

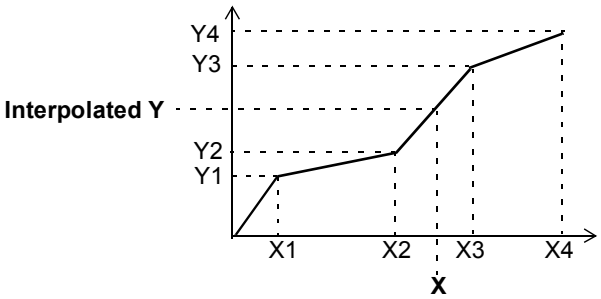
Operation

The output (Y) at the value of the input (X) is calculated with linear interpolation from a piecewise linear function.

$Y = Y_k + (X - X_k)(Y_{k+1} - Y_k) / (X_{k+1} - X_k)$

The piecewise linear function is defined by the X and Y vector tables (XTAB and YTAB). For each X-value in the XTAB table, there is a corresponding Y-value in the YTAB table. The values in XTAB and YTAB must be in ascending order (i.e. from low to high).

XTAB and YTAB values are defined with the DriveSPC tool.



X table (XTAB)	Y table (YTAB)
X1	Y1
X2	Y2
X3	Y3
…	…
X9	Y9

The balancing function (BAL) permits the output signal to track an external reference and gives a smooth return to the normal operation. If BAL is set to 1, output Y is set to the value of the balance reference input (BALREF). The X value which corresponds to this Y value is calculated with linear interpolation and it is indicated by the balance reference output (BALREFO).

If the X input is outside the range defined by the XTAB table, the output Y is set to the highest or lowest value in the YTAB table.

If BALREF is outside the range defined by the YTAB table when balancing is activated (BAL: 0 -> 1), the output Y is set to the value of the BALREF input and the BALREFO output is set to the highest or lowest value in the XTAB table.

The ERROR output is set to 1 when the number of the XTAB and YTAB inputs are different. When ERROR is 1, the FUNG-1V block will not function. XTAB and YTAB tables can be defined in the DATA CONTAINER block (page 285) or the REG-G block (page 292).

Inputs

The input data type is selected by the user.

Balance input (BAL): Boolean

Balance reference input (BALREF): DINT, INT, REAL, REAL24.

X value input (X): DINT, INT, REAL, REAL24

X table input (XTAB): DINT, INT, REAL, REAL24

Y table input (YTAB): DINT, INT, REAL, REAL24

Outputs

Y value output (Y): DINT, INT, REAL, REAL24

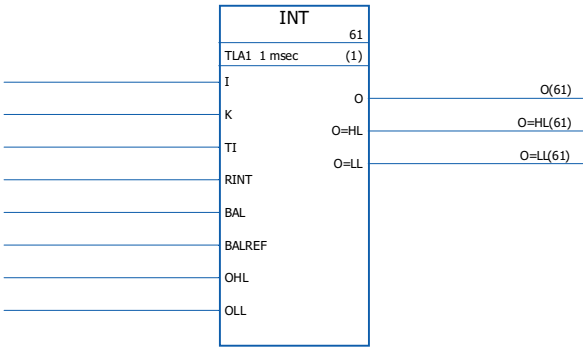
Balance reference output (BALREFO): DINT, INT, REAL, REAL24

Error output (ERROR): Boolean

INT

(10065)

Illustration



Execution time

4.73 µs

Operation

The output (O) is the integrated value of the input (I):

$O(t) = K/TI (\int I(t)\, dt)$

Where TI is the integration time constant and K is the integration gain.

The step response for the integration is:

$O(t) = K \times I(t) \times t/TI$

The transfer function for the integration is:

$G(s) = K\ 1/sTI$

The output value is limited according to the defined minimum and maximum limits (OLL and OHL). If the value is below the minimum value, output O = LL is set to 1. If the value exceeds the maximum value, output O = HL is set to 1. The output (O) retains its value when the input signal I(t) = 0.

The integration time constant is limited to value 2147483 ms. If the time constant is negative, zero time constant is used.

If the ratio between the cycle time and the integration time constant Ts/TI < 1, Ts/TI is set to 1.

The integrator is cleared when the reset input (RINT) is set to 1.

If BAL is set to 1, output O is set to the value of the input BALREF. When BAL is set back to 0, normal integration operation continues.

Inputs

Input (I): REAL

Gain input (K): REAL

Integration time constant input (TI): DINT, 0…2147483 ms

Integrator reset input (RINT): Boolean

Balance input (BAL): Boolean

Balance reference input (BALREF): REAL

Output high limit input (OHL): REAL

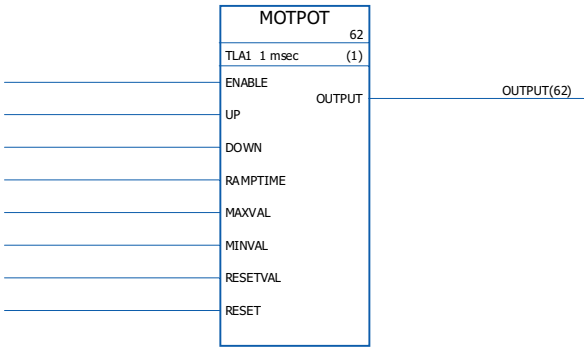
Output low limit input (OLL): REAL

Outputs Output (O): REAL
High limit output (O=HL): Boolean
Low limit output (O=LL): Boolean

MOTPOT

(10067)

Illustration



Execution time 2.92 µs

Operation The motor potentiometer function controls the rate of change of the output from the minimum to the maximum value and vice versa.

The function is enabled by setting the ENABLE input to 1. If the up input (UP) is 1, the output reference (OUTPUT) is increased to the maximum value (MAXVAL) with the defined ramp time (RAMPTIME). If the down input (DOWN) is 1, the output value is decreased to the minimum value (MINVAL) with the defined ramp time. If the up and down inputs are activated/deactivated simultaneously, the output value is not increased/decreased.

If the RESET input is 1, the output will be reset to the value defined by the reset value input (RESETVAL) or to the value defined by the minimum input (MINVAL), whichever is higher.

If the ENABLE input is 0, the output is zero.

Digital inputs are normally used as up and down inputs.

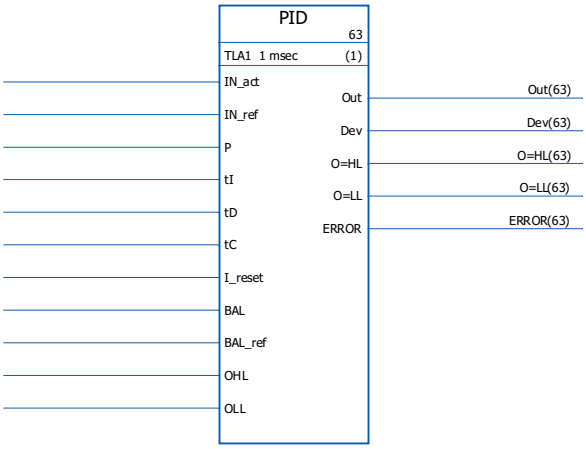
Inputs Function enable input (ENABLE): Boolean
Up input (UP): Boolean
Down input (DOWN): Boolean
Ramp time input (RAMPTIME): REAL (seconds) (i.e. the time required for the output to change from the minimum to the maximum value or from the maximum to the minimum value)
Maximum reference input (MAXVAL): REAL
Minimum reference input (MINVAL): REAL
Reset value input (RESETVAL): REAL
Reset input (RESET): Boolean

Outputs Output (OUTPUT) REAL

PID

(10075)

Illustration



Execution time 15.75 µs

Operation

The PID controller can be used for closed-loop control systems. The controller includes anti-windup correction and output limitation.

The PID controller output (Out) before limitation is the sum of the proportional (U_P), integral (U_I) and derivative (U_D) terms:

$Out_{unlimited}(t) = U_P(t) + U_I(t) + U_D(t)$

$U_P(t) = P \times Dev(t)$

$U_I(t) = P/tI \times [\int Dev(\tau)d\tau + tC \times (Out(t) - Out_{unlimited}(t))]$

$U_D(t) = P \times tD \times d(Dev(t))/dt$

Integrator:

The integral term can be cleared by setting I_reset to 1. Note that the anti-windup correction is simultaneously disabled. When I_reset is 1, the controller acts as a PD controller.

If integration time constant tI is 0, the integral term will not be updated.

Smooth return to normal operation is guaranteed after errors or abrupt input value changes. This is achieved by adjusting the integral term so that the output will retain its previous value during these situations.

Limitation:

The output is limited by the defined minimum and maximum values, OLL and OHL:

If the actual value of the output reaches the specified minimum limit, output O=LL is set to 1.

If the actual value of the output reaches the specified maximum limit, output O=HL is set to 1.

Smooth return to normal operation after limitation is requested if and only if the anti-windup correction is not used, i.e. when tI = 0 or tC = 0.

Error codes:

Error codes are indicated by the error output (ERROR) as follows

Error code	Description
1	The minimum limit (OLL) exceeds the maximum limit (OHL).
2	Overflow with Up, Ui, or Ud calculation

Balancing:

The balancing function (BAL) permits the output signal to track an external reference and gives a smooth return to the normal operation. If BAL is set to 1, the output (Out) is set to the value of the balance reference input (BAL_ref). Balance reference is limited by the defined minimum and maximum limits (OLL and OHL).

Anti-windup:

Anti-windup correction time constant is defined by input tC, which defines the time after which the difference between the unlimited and limited outputs is subtracted from the I-term during limitation. If tC = 0 or tI = 0, anti-windup correction is disabled.

Inputs

Actual input (IN_act): REAL
Reference input (IN_ref): REAL
Proportional gain input (P): REAL
Integration time constant input (tI): REAL. 1 = 1 ms
Derivation time constant input (tD): REAL. 1 = 1 ms
Antiwind-up correction time constant input (tC): IQ6. 1 = 1 ms
Integrator reset input (I_reset): Boolean
Balance input (BAL): Boolean
Balance reference input (BAL_ref): REAL
Output high limit input (OHL): REAL
Output low limit input (OLL): REAL

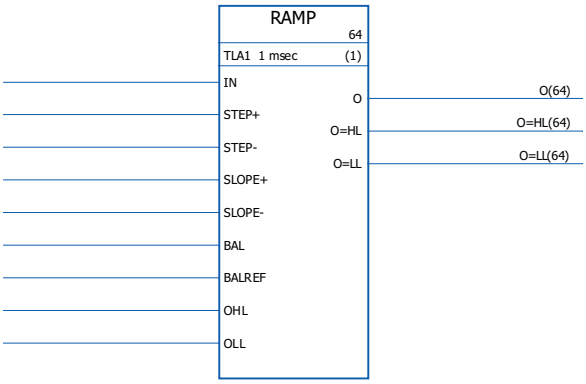
Outputs

Output (Out): REAL
Deviation output (Dev): REAL (= actual -reference = IN_act - IN_ref)
High limit output (O=HL): Boolean
Low limit output (O=LL): Boolean
Error code output (ERROR): INT32

RAMP

(10066)

Illustration



Execution time

4.23 µs

Operation

Limits the rate of the change of the signal.

The input signal (IN) is connected directly to the output (O) if the input signal does not exceed the defined step change limits (STEP+ and STEP-). If the input signal change exceeds these limits, the output signal change is limited by the maximum step change (STEP+/STEP- depending on the direction of rotation). After this, the output signal is accelerated/decelerated by the defined ramp value (SLOPE+/SLOPE-) per second until the input and output signal values are equal.

The output is limited by the defined minimum and maximum values (OLL and OHL). If the actual value of the output falls below the specified minimum limit (OLL), output O=LL is set to 1. If the actual value of the output exceeds the specified maximum limit (OHL), output O=HL is set to 1.

If the balancing input (BAL) is set to 1, the output (O) is set to the value of the balance reference input (BAL_ref). Balancing reference is also limited by the minimum and maximum values (OLL and OHL).

Inputs

Input (IN): REAL
Maximum positive step change input (STEP+): REAL
Maximum negative step change input (STEP-): REAL
Ramp-up value per second input (SLOPE+): REAL
Ramp-down value per second input (SLOPE-): REAL
Balance input (BAL): Boolean
Balance reference input (BALREF): REAL
Output high limit input (OHL): REAL
Output low limit input (OLL): REAL

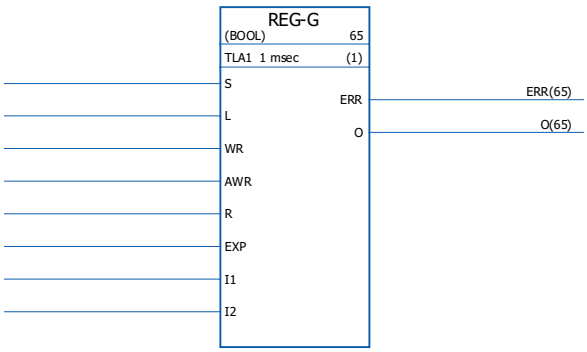
Outputs

Output (O): REAL
High limit output (O=HL): Boolean
Low limit output (O=LL): Boolean

REG-G

(10102)

Illustration



Execution time

-

Operation

Combines the array (group of variables) (if any) on the EXP input with the values of the I1…I32 pins to produce an output array. The data type of the arrays can be INT, DINT, REAL16, REAL24 or Boolean. The output array consists of the data from the EXP input and the values of the I1…In (in this order).

When input S is 1, data is continuously assembled into the output array. The element acts as a latch when input S is 0; the latest data assembled then remains at the output.

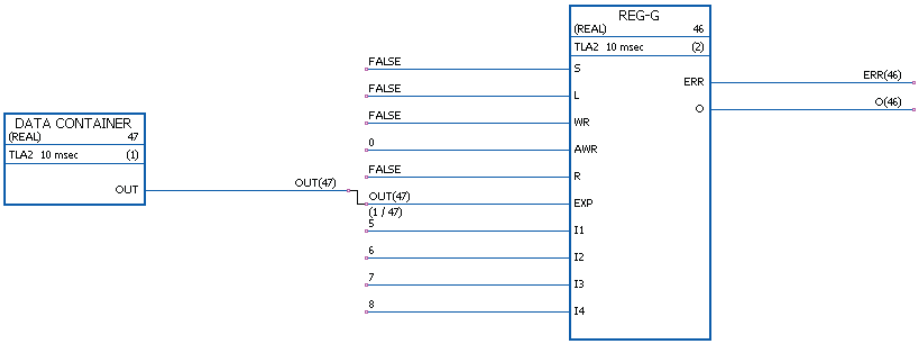
If S is 0 and L changes state from 0 to 1, the array from the EXP input and the values of the I1…In inputs are copied to output O during this program cycle. If S or R is 1, L has no effect.

WR and AWR are used to change individual cells of the output array. AWR indicates the input whose value is moved to the output array. If AWR is 0, only the array from input EXP is moved to the output. If AWR is not 0, the corresponding I input is moved to the output. This is performed when WR goes from 0 to 1.

When input R is 1, the output array is cleared and all further data entry is prevented. R overrides both S and L. If WR is 1, the address at AWR is checked and if it is illegal (negative or greater than the number of inputs), the error output (ERR) is set to 2. Otherwise ERR is 0.

Whenever an error is detected, ERR is set within one cycle. No place in the register is affected when an error occurs.

Example:



In the diagram, the DATA CONTAINER block includes an array with values [1,2,3,4]. At start, the output array is [0,0,0,0,0,0,0,0]. When WR changes to 1 and returns to 0, the AWR value of 0 means that only EXP is moved into the output array, which now reads [1,2,3,4,0,0,0,0]. After this, AWR is changed to 3, meaning that inputs EXP and I3 are moved to the output. After a WR switch, the output array is [1,2,3,4,0,0,7,0].

Inputs

Set (S): Boolean, INT, DINT, REAL, REAL24

Load (L): Boolean, INT, DINT, REAL, REAL24

Write (WR): Boolean, INT, DINT, REAL, REAL24

Write address (AWR): INT

Reset (R): Boolean

Expander (EXP): IArray

Data input (I1…I32): Boolean, INT, DINT, REAL, REAL24

Outputs

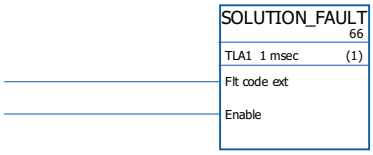
Error (ERR): INT

Array data output (O): OC1

SOLUTION_FAULT

(10097)

Illustration



Execution time -

Operation When the block is enabled (by setting the Enable input to 1), a fault (F-0317 SOLUTION FAULT) is generated by the drive. The value of the Flt code ext input is recorded by the fault logger.

Inputs Fault code extension (Flt code ext): DINT

Generate fault (Enable): Boolean

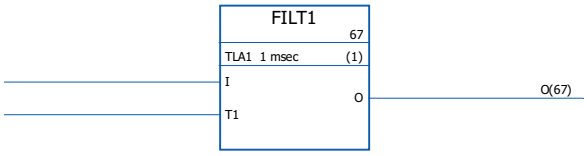
Outputs -

Filters

FILT1

(10069)

Illustration



Execution time 7.59 µs

Operation The output (O) is the filtered value of the input (I) value and the previous output value (O_{prev}). The FILT1 block acts as 1st order low pass filter.

Note: Filter time constant (T1) must be selected so that T1/Ts < 32767. If the ratio exceeds 32767, it is considered as 32767. Ts is the cycle time of the program in ms.

If T1 < Ts, the output value is the input value.

The step response for a single pole low pass filter is:

$O(t) = I(t) \times (1 - e^{-t/T1})$

The transfer function for a single pole low pass filter is:

$G(s) = 1/ (1 + sT1)$

Inputs Input (I): REAL

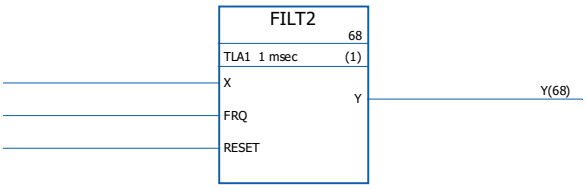
Filter time constant input (T1): DINT, 1 = 1 ms

Outputs Output (O): REAL

FILT2

(10070)

Illustration



Execution time 6.30 µs

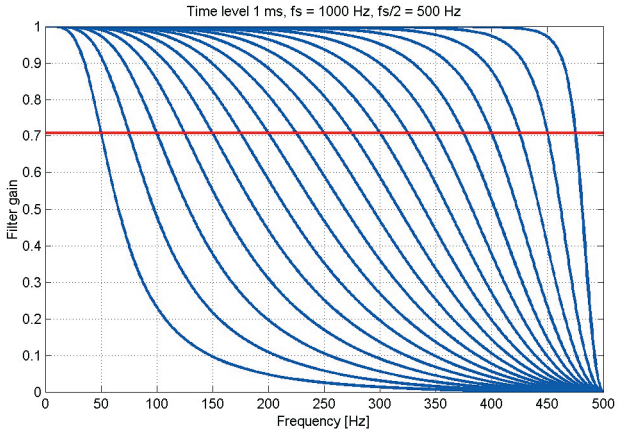
Operation The output (Y) is the filtered value of the input (X). The FILT2 block acts as a 2nd order low pass filter.

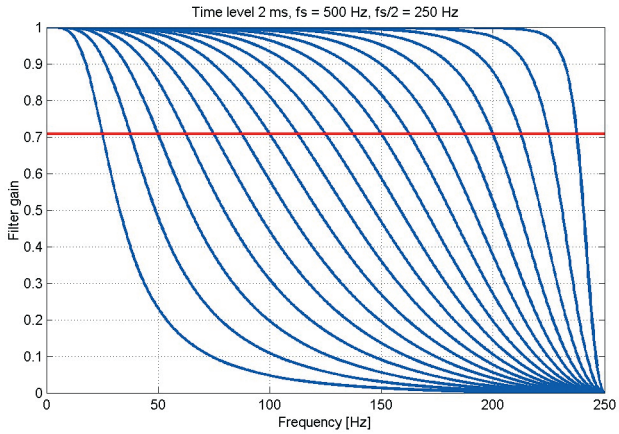
When the RESET input value is set to 1, the input is connected to the output without filtering.

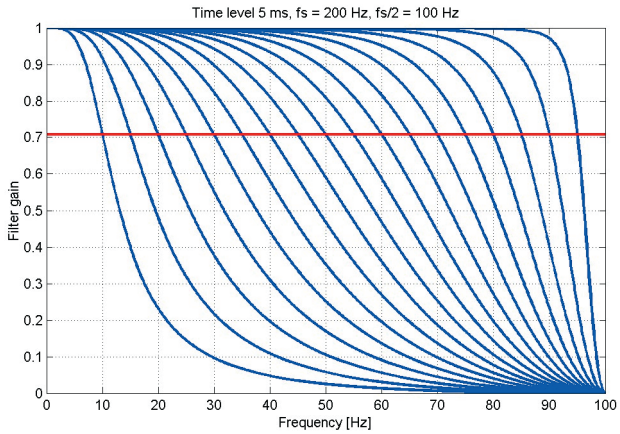
Notes:

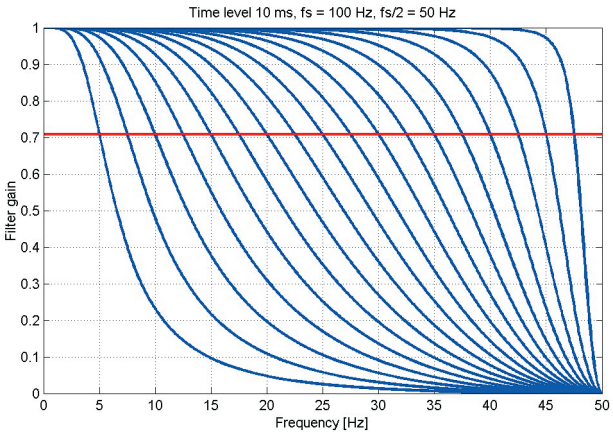
- The -3 dB cutoff frequency (FRQ) is limited to its maximum value (16383 Hz).
- The frequency of the input signal must be less than half of sampling frequency (fs) – any higher frequencies are aliased to the allowable range. The sampling frequency is defined by the time level of the block; for example, 1 ms corresponds to a sampling frequency of 1000 Hz.

The following diagrams show the frequency responses for 1, 2, 5 and 10 ms time levels. The -3 dB cutoff level is represented as the horizontal line at 0.7 gain.


Time level 1 ms, fs = 1000 Hz, fs/2 = 500 Hz
Filter gain
Frequency [Hz]
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300
350
400
450
500


Time level 2 ms, fs = 500 Hz, fs/2 = 250 Hz
Filter gain
Frequency [Hz]
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250


Time level 5 ms, fs = 200 Hz, fs/2 = 100 Hz
Filter gain
Frequency [Hz]
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90
100



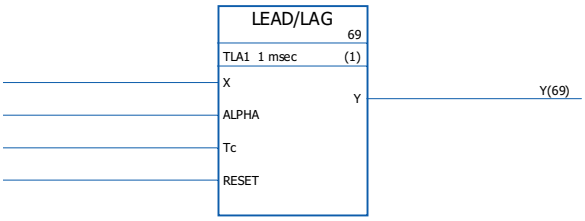
Inputs Input (X): REAL
-3 dB cutoff frequency input (FRQ): DINT (0…16383 Hz)
Reset input (RESET): Boolean

Outputs Output (Y): REAL

LEAD/LAG

(10071)

Illustration



Execution time 5.55 µs

Operation The output (Y) is the filtered value of the input (X). When ALPHA > 1, the function block acts as a lead filter. When ALPHA < 1, the function block acts as a lag filter. When ALPHA = 1, no filtering occurs.

The transfer function for a lead/lag filter is:

$(1 + \text{ALPHA}T_c s) / (1 + T_c s)$

When RESET input is 1, the input value (X) is connected to the output (Y).

If ALPHA or Tc < 0, the negative input value is set to zero before filtering.

Inputs Input (X): REAL
Lead/Lag filter type input (ALPHA): REAL
Time constant input (Tc): REAL
Reset input (RESET): Boolean

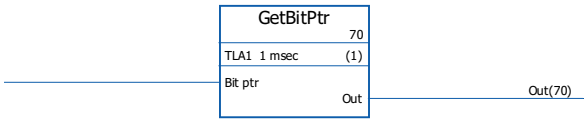
Outputs Output (Y): REAL

Parameters

GetBitPtr

(10099)

Illustration



Execution time -

Operation Reads the status of one bit within a parameter value cyclically.
The Bit ptr input specifies the parameter group, index and bit to be read.
The output (Out) provides the value of the bit.

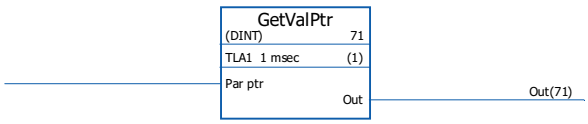
Inputs Parameter group, index and bit (Bit ptr): DINT

Outputs Bit status (Out): DINT

GetValPtr

(10098)

Illustration



Execution time -

Operation Reads the value of a parameter cyclically.
The Par ptr input specifies the parameter group and index to be read.
The output (Out) provides the value of the parameter.

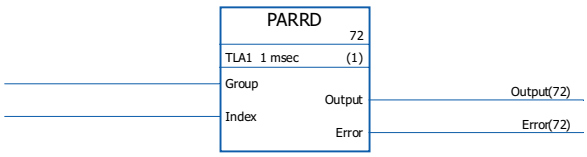
Inputs Parameter group and index (Par ptr): DINT

Outputs Parameter value (Out): DINT

PARRD

(10082)

Illustration



Execution time 6.00 µs

Operation

Reads the scaled value of a parameter (specified by the Group and Index inputs). If the parameter is a pointer parameter, the Output pin provides the number of the source parameter instead of its value.

Error codes are indicated by the error output (Error) as follows:

Error code	Description
0	No error
<> 0	Error

See also blocks PARRDINTR and PARRDPTR.

Inputs

Parameter group input (Group): DINT

Parameter index input (Index): DINT

Outputs

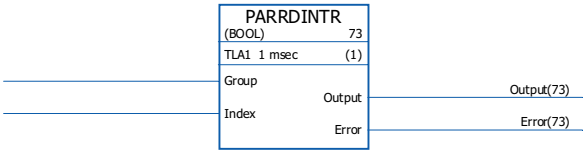
Output (Output): DINT

Error output (Error): DINT

PARRDINTR

(10101)

Illustration



Execution time

-

Operation

Reads the internal (non-scaled) value of a parameter (specified by the Group and Index inputs). The value is provided by the Output pin.

Error codes are indicated by the error output (Error) as follows:

Error code	Description
0	No error or busy
<> 0	Error

Note: Using this block may cause incompatibility issues when upgrading the application to another firmware version.

Inputs

Parameter group (Group): DINT

Parameter index (Index): DINT

Outputs

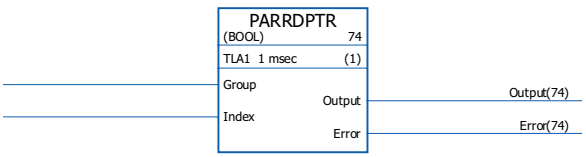
Output (Output): Boolean, INT, DINT, REAL, REAL24

Error output (Error): DINT

PARRDPTR

(10100)

Illustration



Execution time

-

Operation Reads the internal (non-scaled) value of the source of a pointer parameter. The pointer parameter is specified using the Group and Index inputs.

The value of the source selected by the pointer parameter is provided by the Output pin.

Error codes are indicated by the error output (Error) as follows:

Error code	Description
0	No error or busy
<> 0	Error

Inputs Parameter group (Group): DINT

Parameter index (Index): DINT

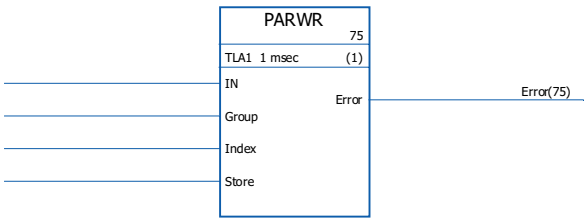
Outputs Output (Output): Boolean, INT, DINT, REAL, REAL24

Error output (Error): DINT

PARWR

(10080)

Illustration



Execution time 14.50 µs

Operation The input value (IN) is written to the defined parameter (Group and Index).

The new parameter value is stored to the flash memory if the store input (Store) is 1.
Note: Cyclic parameter value storing can damage the memory unit. Parameter values should be stored only when necessary.

Error codes are indicated by the error output (Error) as follows:

Error code	Description
0	No error
<> 0	Error

Inputs Input (IN): DINT

Parameter group input (Group): DINT

Parameter index input (Index): DINT

Store input (Store): Boolean

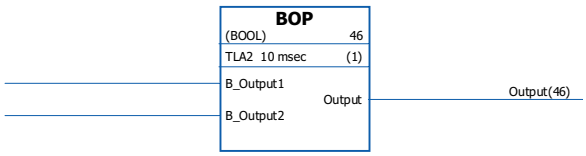
Outputs Error output (Error): DINT

Program structure

BOP

(10105)

Illustration



Execution time -

Operation The BOP (Bundle OutPut) block collects the outputs of several different sources. The sources are connected to the B_Output pins. The B_Output pin that changed last is relayed to the Output pin.

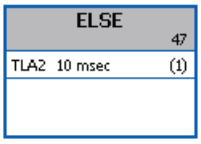
The block is intended for use with conditional IF-ENDIF structures. See the example under the IF block.

Inputs Values from different conditional branches (B_Output1…B_OutputN): INT, DINT, Boolean, REAL, REAL24

Outputs Output from currently active branch of a IF-ELSEIF structure or latest updated input value (Output): INT, DINT, Boolean, REAL, REAL24

ELSE

Illustration



Execution time -

Operation See description of IF block.

Inputs -

Outputs -

ELSEIF

Illustration

Execution time -

Operation See description of IF block.

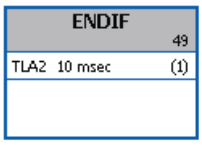
Inputs -

Inputs Input (COND): Boolean

Outputs -

ENDIF

Illustration



Execution time -

Operation See description of IF block.

Inputs -

Outputs -

IF

(10103)

Illustration

Execution time -

Operation

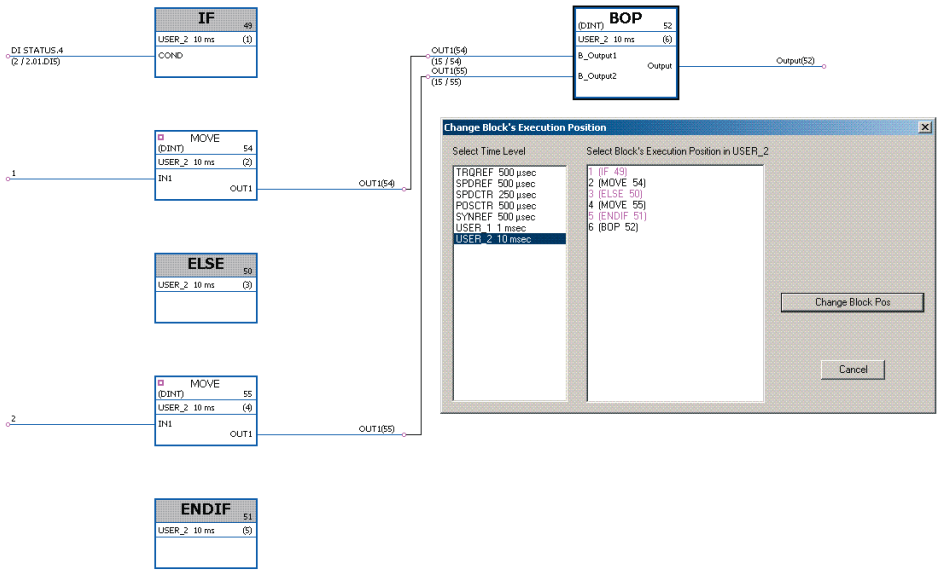
The IF, ELSE, ELSEIF and ENDIF blocks define, by Boolean logic, which parts of the application program are executed.

If the condition input (COND) is true, the blocks between the IF block and the next ELSEIF, ELSE or ENDIF block (in execution order) are run. If the condition input (COND) is false, the blocks between the IF block and the next ELSEIF, ELSE or ENDIF block are skipped.

The outputs of the “branches” are collected and selected by using the BOP block.

Example:

Bit 4 of 2.01 DI STATUS (digital input DI5) controls the branching of the application program. If the input is 0, the blocks between the IF and ELSE blocks are skipped but the blocks between ELSE and ENDIF are run. If the input is 1, the blocks between IF and ELSE are run. The program execution then jumps to the block that follows ENDIF, which is a BOP. The BOP block outputs the value from the branch that was executed. If the digital input is 0, the BOP block output is 2; if the digital input is 1, the BOP block output is 1.



Inputs

Input (COND): Boolean

Outputs

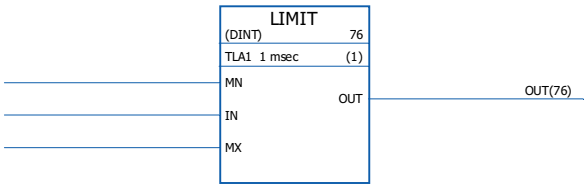
-

Selection

LIMIT

(10052)

Illustration



Execution time 0.53 µs

Operation The output (OUT) is the limited input (IN) value. Input is limited according to the minimum (MN) and maximum (MX) values.

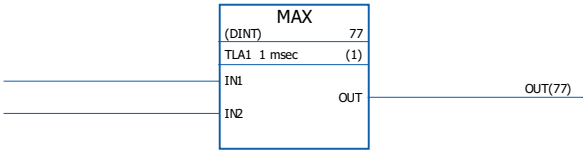
Inputs The input data type is selected by the user.
Minimum input limit (MN): INT, DINT, REAL, REAL24
Input (IN): INT, DINT, REAL, REAL24
Maximum input limit (MX): INT, DINT, REAL, REAL24

Outputs Output (OUT): INT, DINT, REAL, REAL24

MAX

(10053)

Illustration



Execution time 0.81 µs (when two inputs are used) + 0.53 µs (for every additional input). When all inputs are used, the execution time is 16.73 µs.

Operation The output (OUT) is the highest input value (IN).

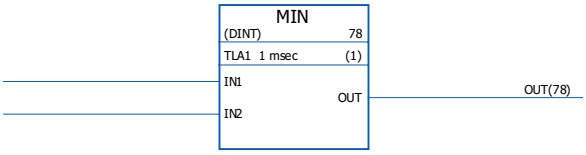
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): INT, DINT, REAL, REAL24

MIN

(10054)

Illustration



Execution time 0.81 µs (when two inputs are used) + 0.52 µs (for every additional input). When all inputs are used, the execution time is 16.50 µs.

Operation The output (OUT) is the lowest input value (IN).

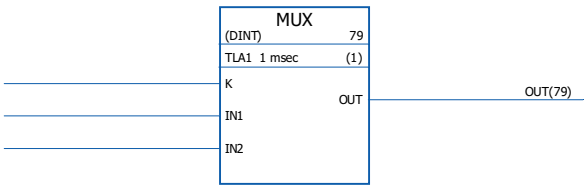
Inputs The input data type and the number of inputs (2…32) are selected by the user.
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): INT, DINT, REAL, REAL24

MUX

(10055)

Illustration



Execution time 0.70 µs

Operation The value of an input (IN) selected by the address input (K) is stored to the output (OUT).
If the address input is 0, negative or exceeds the number of the inputs, the output is 0.

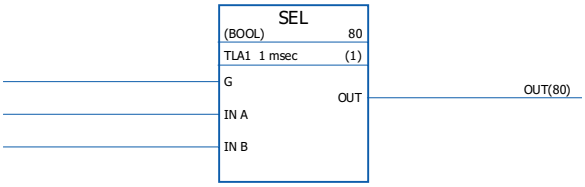
Inputs The input data type and number of inputs (2…32) are selected by the user.
Address input (K): DINT
Input (IN1…IN32): INT, DINT, REAL, REAL24

Outputs Output (OUT): INT, DINT, REAL, REAL24

SEL

(10056)

Illustration



Execution time 1.53 µs

Operation The output (OUT) is the value of the input (IN) selected by the selection input (G).
If G = 0: OUT = IN A.
If G = 1: OUT = IN B.

Inputs The input data type is selected by the user.
Selection input (G): Boolean
Input (IN A, IN B): Boolean, INT, DINT, REAL, REAL24

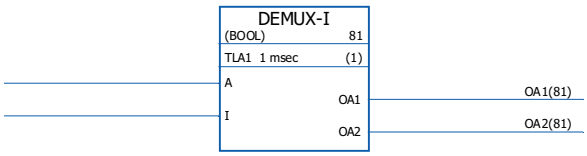
Outputs Output (OUT): Boolean, INT, DINT, REAL, REAL24

Switch & Demux

DEMUX-I

(10061)

Illustration



Execution time 1.38 µs (when two outputs are used) + 0.30 µs (for every additional output). When all outputs are used, the execution time is 10.38 µs.

Operation Input (I) value is stored to the output (OA1…OA32) selected by the address input (A). All other outputs are 0.

If the address input is 0, negative or exceeds the number of the outputs, all outputs are 0.

Inputs The input data type is selected by the user.

Address input (A): DINT

Input (I): INT, DINT, Boolean, REAL, REAL24

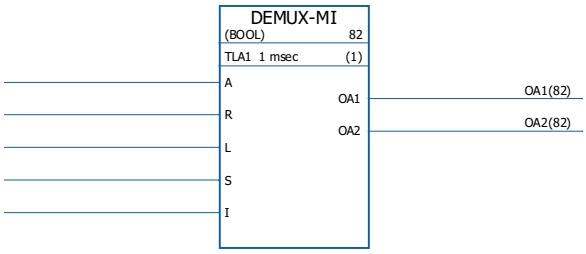
Outputs The number of the output channels (1…32) is selected by the user.

Output (OA1…OA32): INT, DINT, REAL, REAL24, Boolean

DEMUX-MI

(10062)

Illustration



Execution time 0.99 µs (when two outputs are used) + 0.25 µs (for every additional output). When all outputs are used, the execution time is 8.4 µs.

Operation

The input (I) value is stored to the output (OA1…OA32) selected by the address input (A) if the load input (L) or the set input (S) is 1. When the load input is set to 1, the input (I) value is stored to the output only once. When the set input is set to 1, the input (I) value is stored to the output every time the block is executed. The set input overrides the load input.

If the reset input (R) is 1, all connected outputs are 0.

If the address input is 0, negative or exceeds the number of the outputs, all outputs are 0.

Example:

S	L	R	A	I	OA1	OA2	OA3	OA4
1	0	0	2	150	0	150	0	0
0	0	0	2	120	0	150	0	0
0	1	0	3	100	0	150	100	0
1	0	0	1	200	200	150	100	0
1	1	0	4	250	200	150	100	250
1	1	1	2	300	0	0	0	0

Inputs

The input data type is selected by the user.

Address input (A): DINT

Reset input (R): Boolean

Load input (L): Boolean

Set input (S): Boolean

Input (I): DINT, INT, REAL, REAL24, Boolean

Outputs

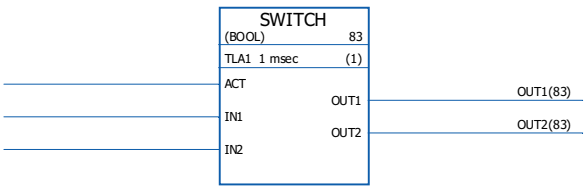
The number of the output channels (1…32) is selected by the user.

Output (OA1…OA32): DINT, INT, REAL, REAL24, Boolean

SWITCH

(10063)

Illustration



Execution time

0.68 µs (when two inputs are used) + 0.50 µs (for every additional input). When all inputs are used, the execution time is 15.80 µs.

Operation

The output (OUT) is equal to the corresponding input (IN) if the activate input (ACT) is 1. Otherwise the output is 0.

Inputs

The input data type and the number of inputs (1…32) are selected by the user.

Activate input (ACT): Boolean

Input (IN1…IN32): INT, DINT, REAL, REAL24, Boolean

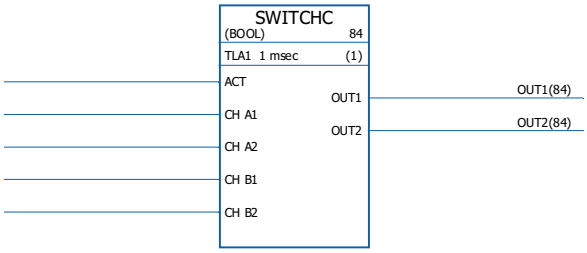
Outputs

Output (OUT1…OUT32): INT, DINT, REAL, REAL24, Boolean

SWITCHC

(10064)

Illustration



Execution time 1.53 µs (when two inputs are used) + 0.73 µs (for every additional input). When all inputs are used, the execution time is 23.31 µs.

Operation The output (OUT) is equal to the corresponding channel A input (CH A1…32) if the activate input (ACT) is 0. The output is equal to the corresponding channel B input (CH B1…32) if the activate input (ACT) is 1.

Inputs The input data type and the number of inputs (1…32) are selected by the user.

Activate input (ACT): Boolean

Input (CH A1…CH A32, CH B1…CH B32): INT, DINT, REAL, REAL24, Boolean

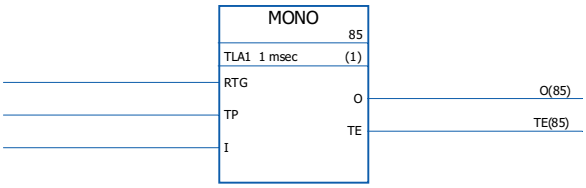
Outputs Output (OUT1…OUT32): INT, DINT, REAL, REAL24, Boolean

Timers

MONO

(10057)

Illustration



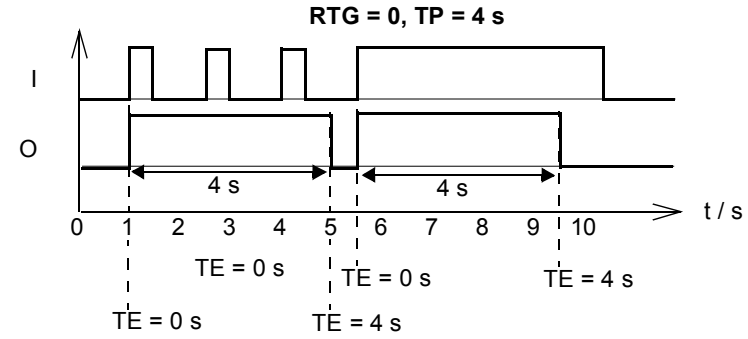
Execution time 1.46 µs

Operation The output (O) is set to 1 and the timer is started, if the input (I) is set to 1. The output is reset to 0 when the time defined by the time pulse input (TP) has elapsed. Elapsed time (TE) count starts when the output is set to 1 and stops when the output is set to 0.

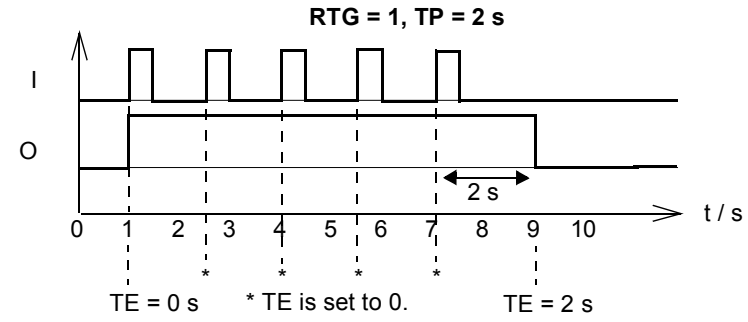
If RTG is 0, a new input pulse during the time defined by TP has no effect on the function. The function can be restarted only after the time defined by TP has elapsed.

If RTG is 1, a new input pulse during the time defined by TP restarts the timer and sets the elapsed time (TE) to 0.

Example 1: MONO is not re-triggable, i.e. RTG = 0.



Example 2: MONO is re-triggable, i.e. RTG = 1.



Inputs Re-trigger input (RTG): Boolean

Time pulse input (TP): DINT (1 = µs)

Input (I): Boolean

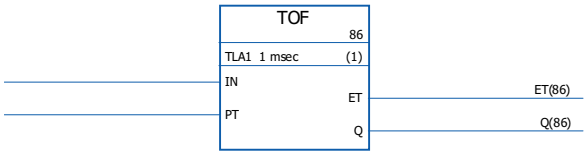
Outputs Output (O): Boolean

Time elapsed output (TE): DINT (1 = 1 µs)

TOF

(10058)

Illustration

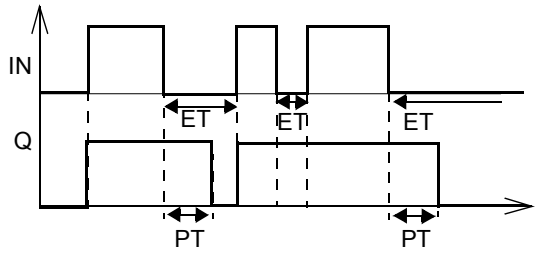


Execution time 1.10 µs

Operation The output (Q) is set to 1, when the input (IN) is set to 1. The output is reset to zero when the input has been 0 for a time defined by the pulse time input (PT).

Elapsed time count (ET) starts when the input is set to 0 and stops when the input is set to 1.

Example:



Inputs Input (IN): Boolean

Pulse time input (PT): DINT (1 = 1 µs)

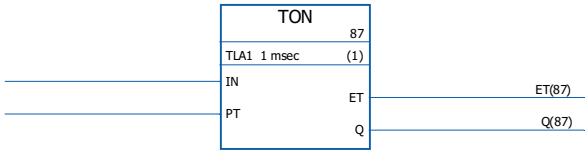
Outputs Elapsed time output (ET): DINT (1 = 1 µs)

Output (Q): Boolean

TON

(10059)

Illustration



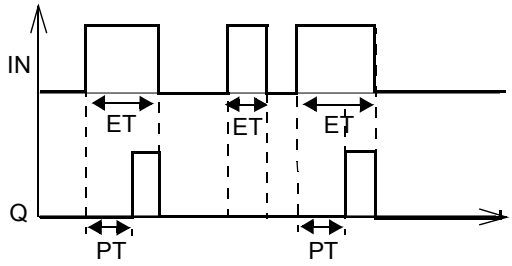
Execution time 1.22 µs

Operation

The output (Q) is set to 1 when the input (IN) has been 1 for a time defined by the pulse time input (PT). The output is set to 0, when the input is set to 0.

Elapsed time count (ET) starts when the input is set to 1 and stops when the input is set to 0.

Example:



Inputs

Input (IN): Boolean

Pulse time input (PT): DINT (1 = 1 µs)

Outputs

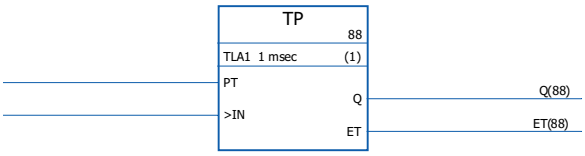
Elapsed time output (ET): DINT (1 = 1 µs)

Output (Q): Boolean

TP

(10060)

Illustration



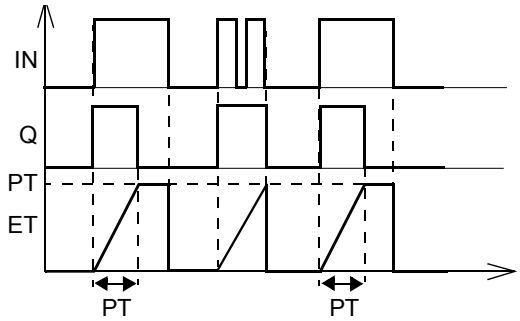
Execution time

1.46 µs

Operation

The output (Q) is set to 1 when the input (IN) is set to 1. The output is set to 0, when it has been 1 for a time defined by the pulse time input (PT).

Elapsed time count (ET) starts when the input is set to 1 and stops when the input is set to 0.



Inputs

Pulse time input (PT): DINT (1 = 1 µs)

Input (IN): Boolean

Outputs

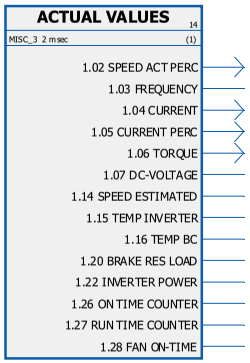
Output (Q): Boolean

Elapsed time output (ET): DINT (1 = 1 µs)

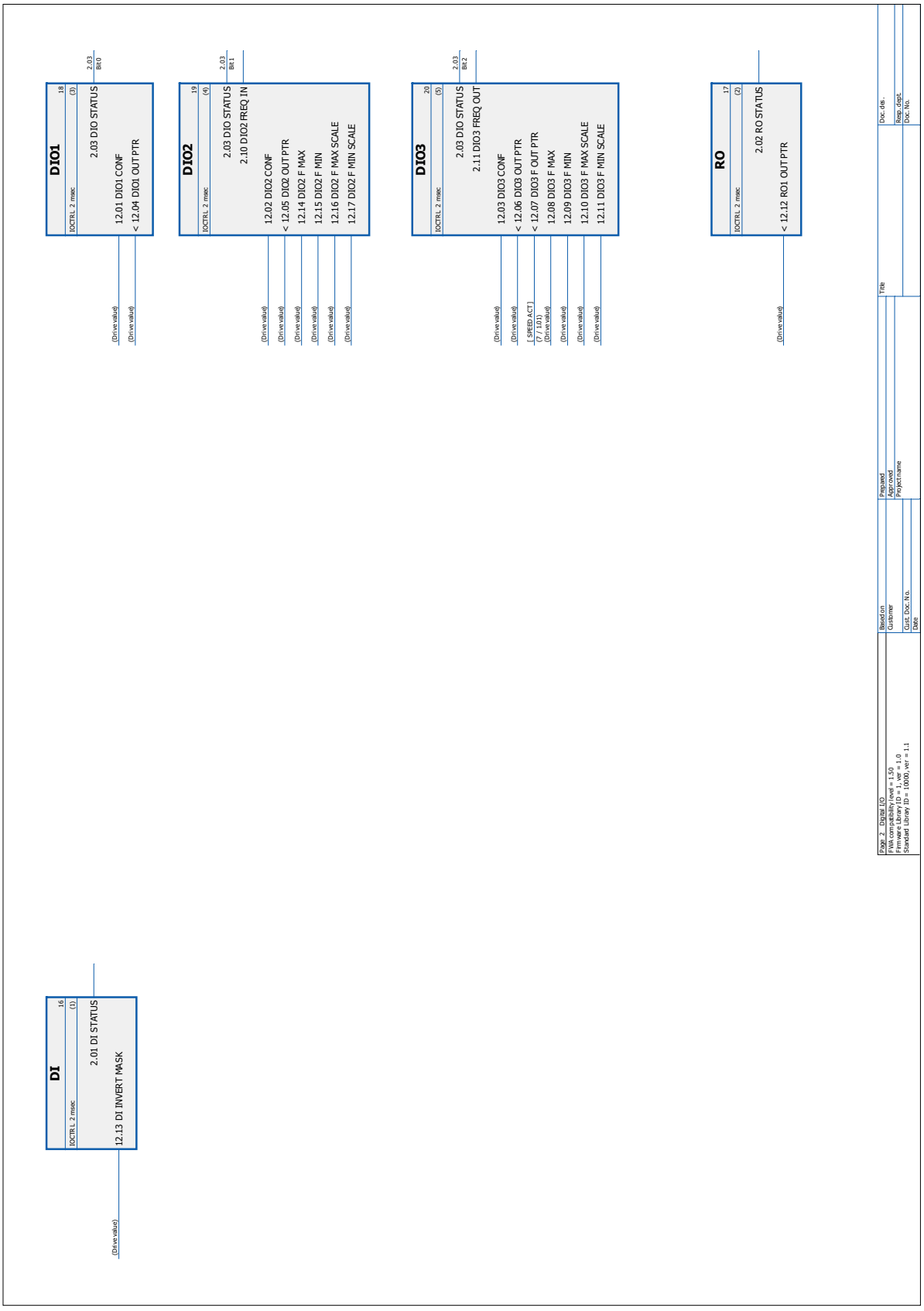
Application program template

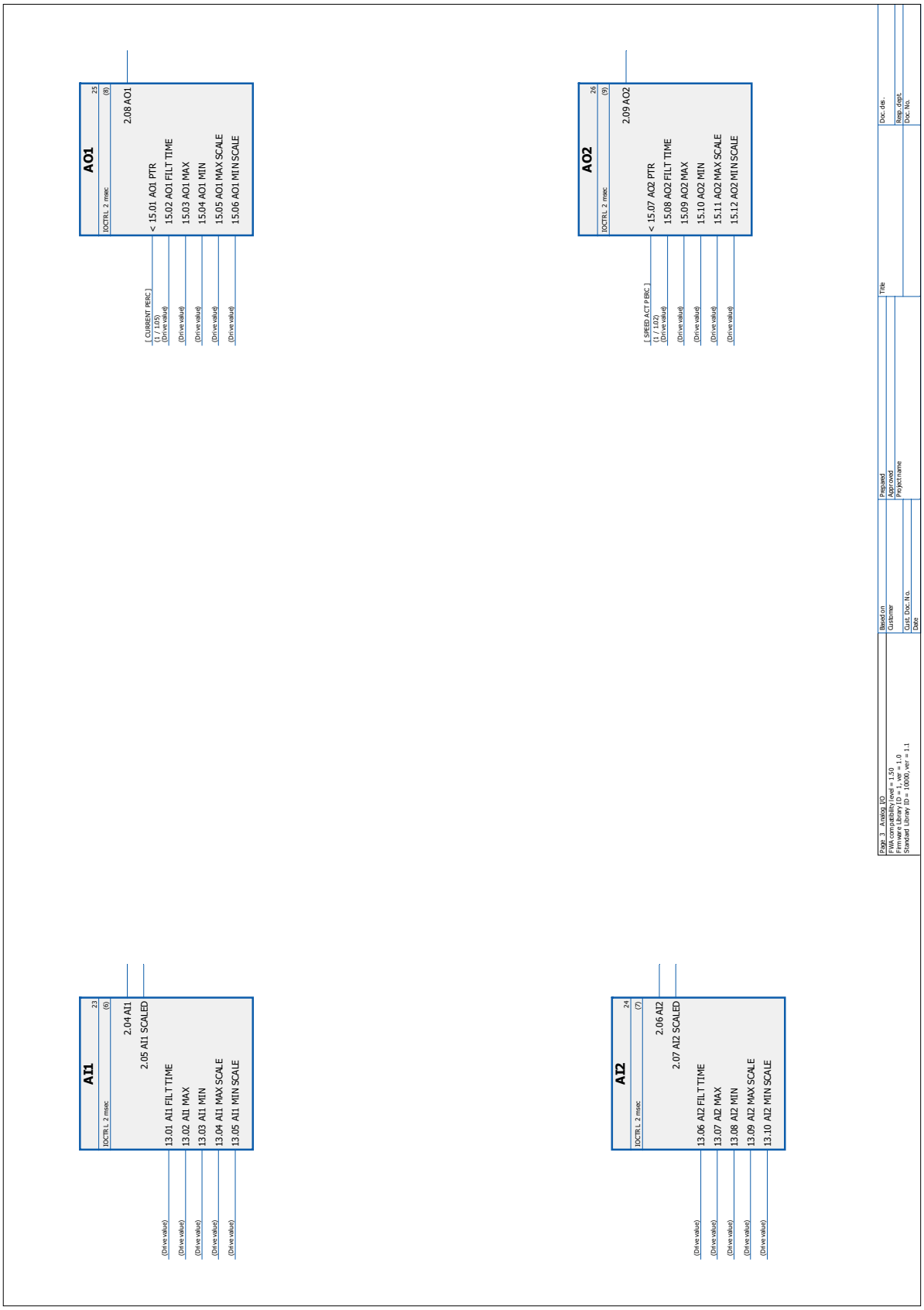
What this chapter contains

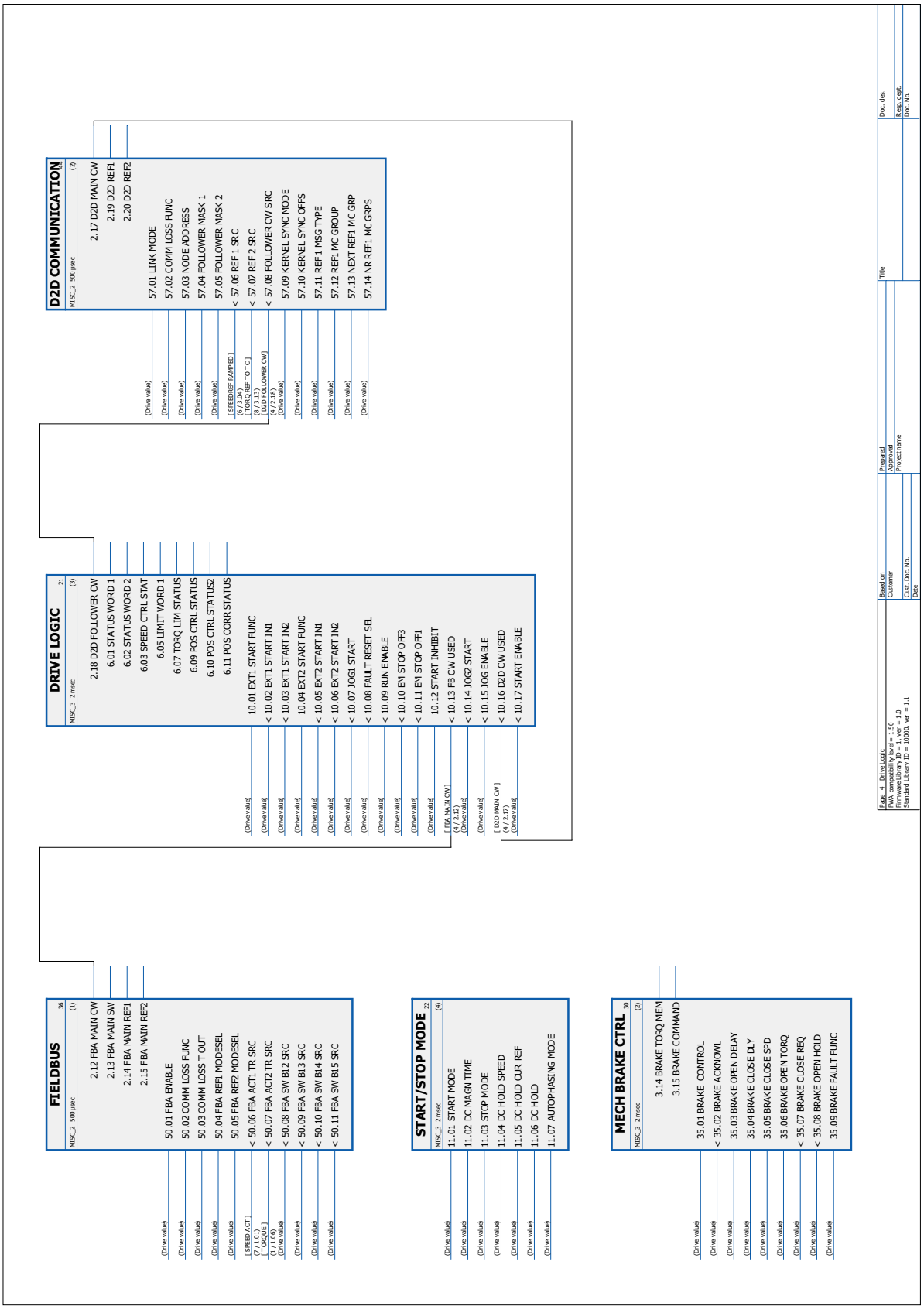
This chapter presents the application program template as displayed by the DriveSPC tool after empty template upload (Drive - Upload Template from Drive).

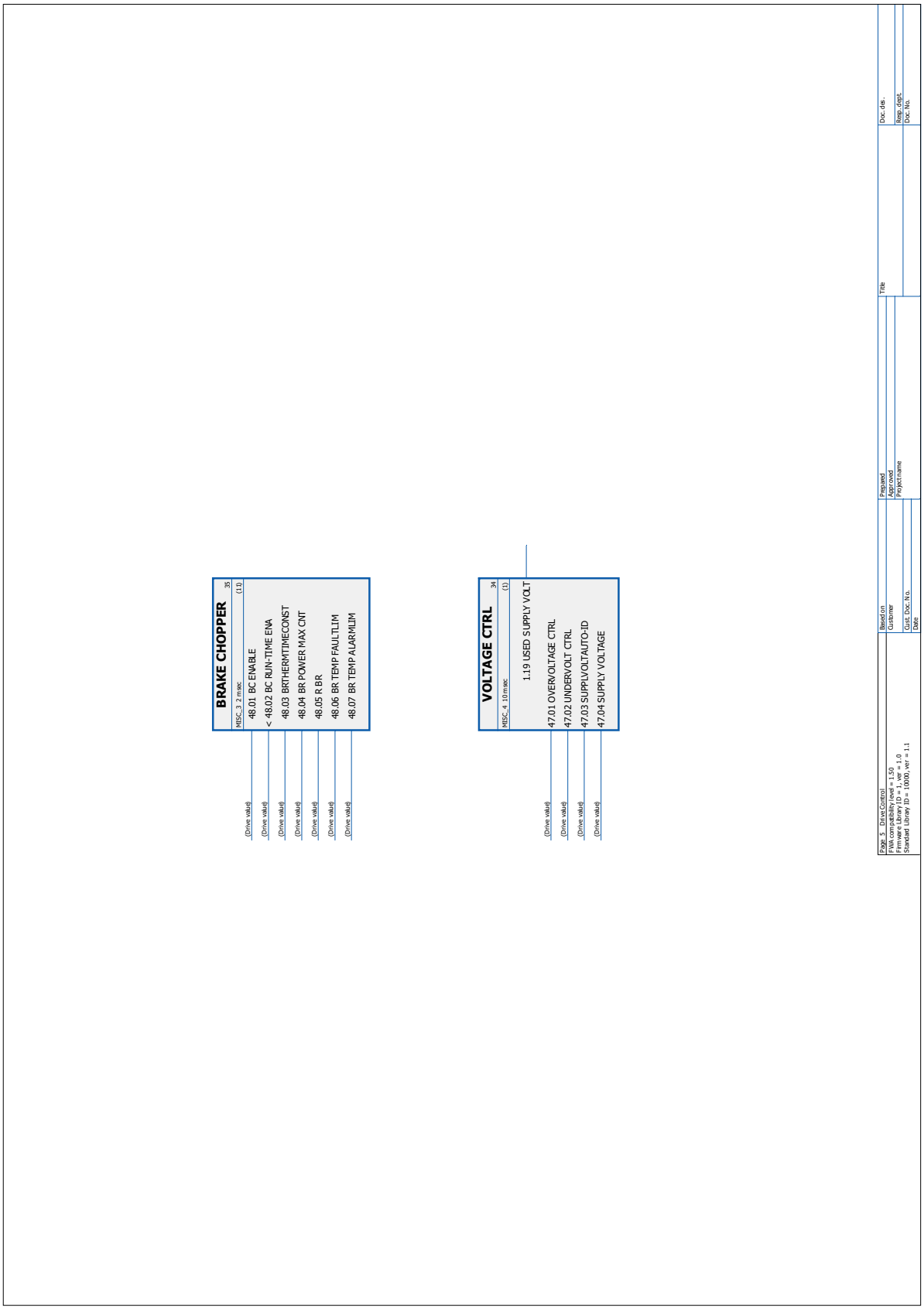


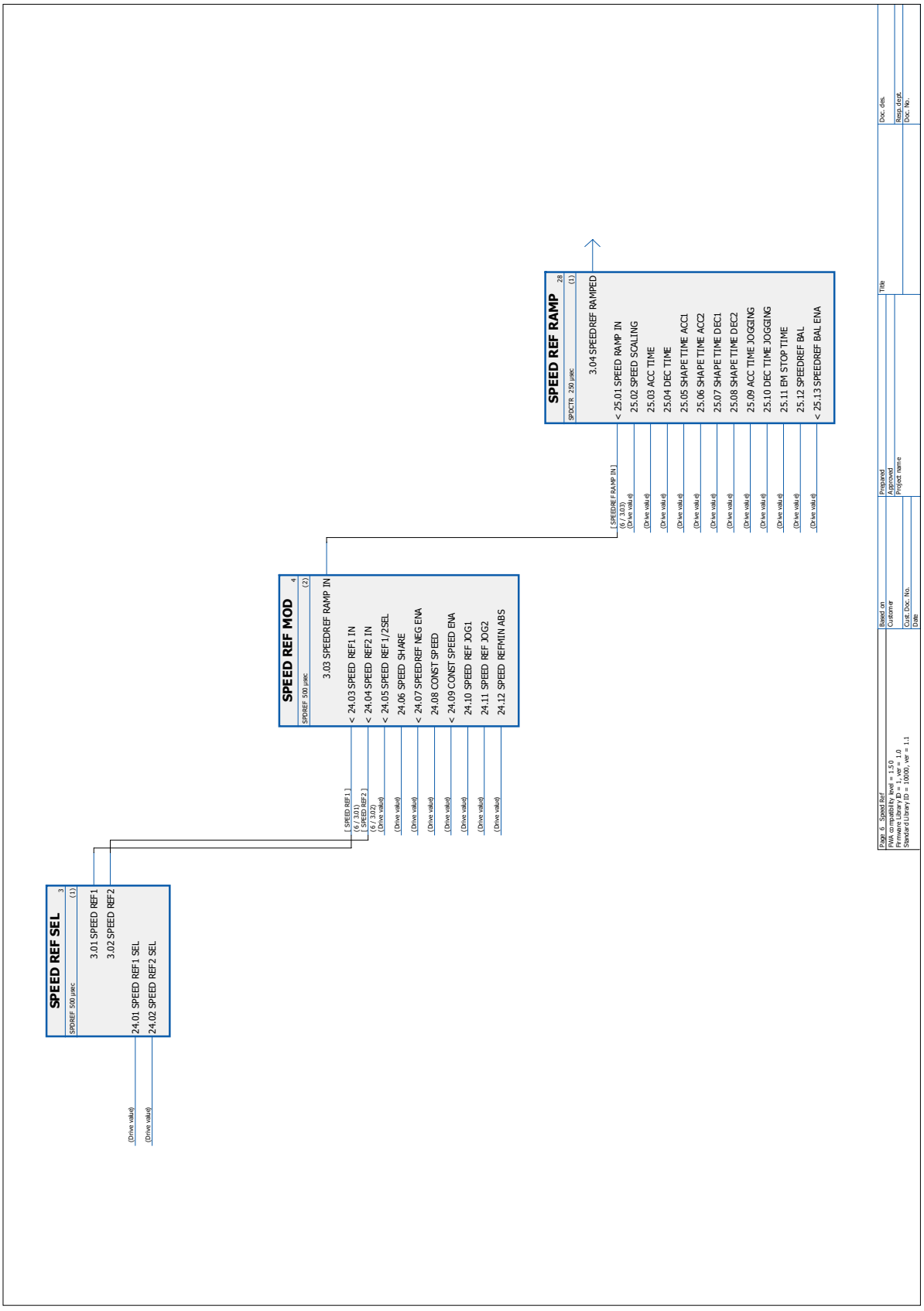
Page 1 Signals	Based on	Prepared	Title	Doc. des.
FWA compatibility level = 1.50	Customer	Approved		
Firmware Library ID = 1, ver = 1.0		Project name		Resp. dept.
Standard Library ID = 10000, ver = 1.1	Cust. Doc. No.			Doc. No.
	Date			

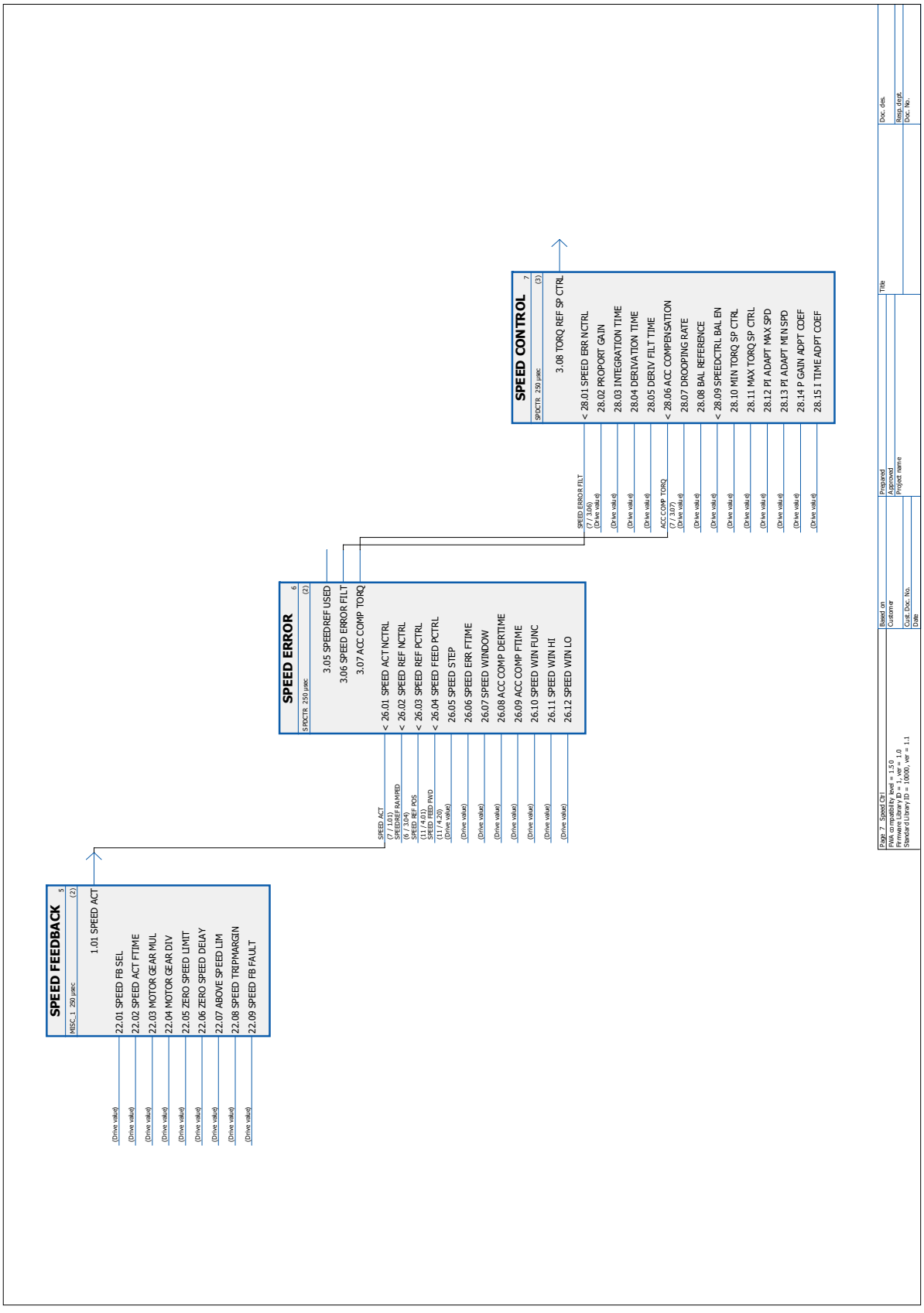

DI
IOCTRL 2 msec
16
(1)
2.01 DI STATUS
12.13 DI INVERT MASK
(Drive value)
DIO1
IOCTRL 2 msec
18
(3)
2.03 DIO STATUS
12.01 DIO1 CONF
< 12.04 DIO1 OUT PTR
2.03 Bit 0
(Drive value)
(Drive value)
DIO2
IOCTRL 2 msec
19
(4)
2.03 DIO STATUS
2.10 DIO2 FREQ IN
12.02 DIO2 CONF
< 12.05 DIO2 OUT PTR
12.14 DIO2 F MAX
12.15 DIO2 F MIN
12.16 DIO2 F MAX SCALE
12.17 DIO2 F MIN SCALE
2.03 Bit 1
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
DIO3
IOCTRL 2 msec
20
(5)
2.03 DIO STATUS
2.11 DIO3 FREQ OUT
12.03 DIO3 CONF
< 12.06 DIO3 OUT PTR
< 12.07 DIO3 F OUT PTR
12.08 DIO3 F MAX
12.09 DIO3 F MIN
12.10 DIO3 F MAX SCALE
12.11 DIO3 F MIN SCALE
2.03 Bit 2
(Drive value)
(Drive value)
[SPEED ACT] (7 / 1.01)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
RO
IOCTRL 2 msec
17
(2)
2.02 RO STATUS
< 12.12 RO1 OUT PTR
(Drive value)
Page 2 Digital I/O
FWA compatibility level = 1.50
Firmware library ID = 1, ver = 1.0
Standard Library ID = 1000, ver = 1.1
Based on
Customer
Cust. Doc. No.
Date
Prepared
Approved
Project name
Title
Doc. dis.
Resp. dept.
Doc. No.

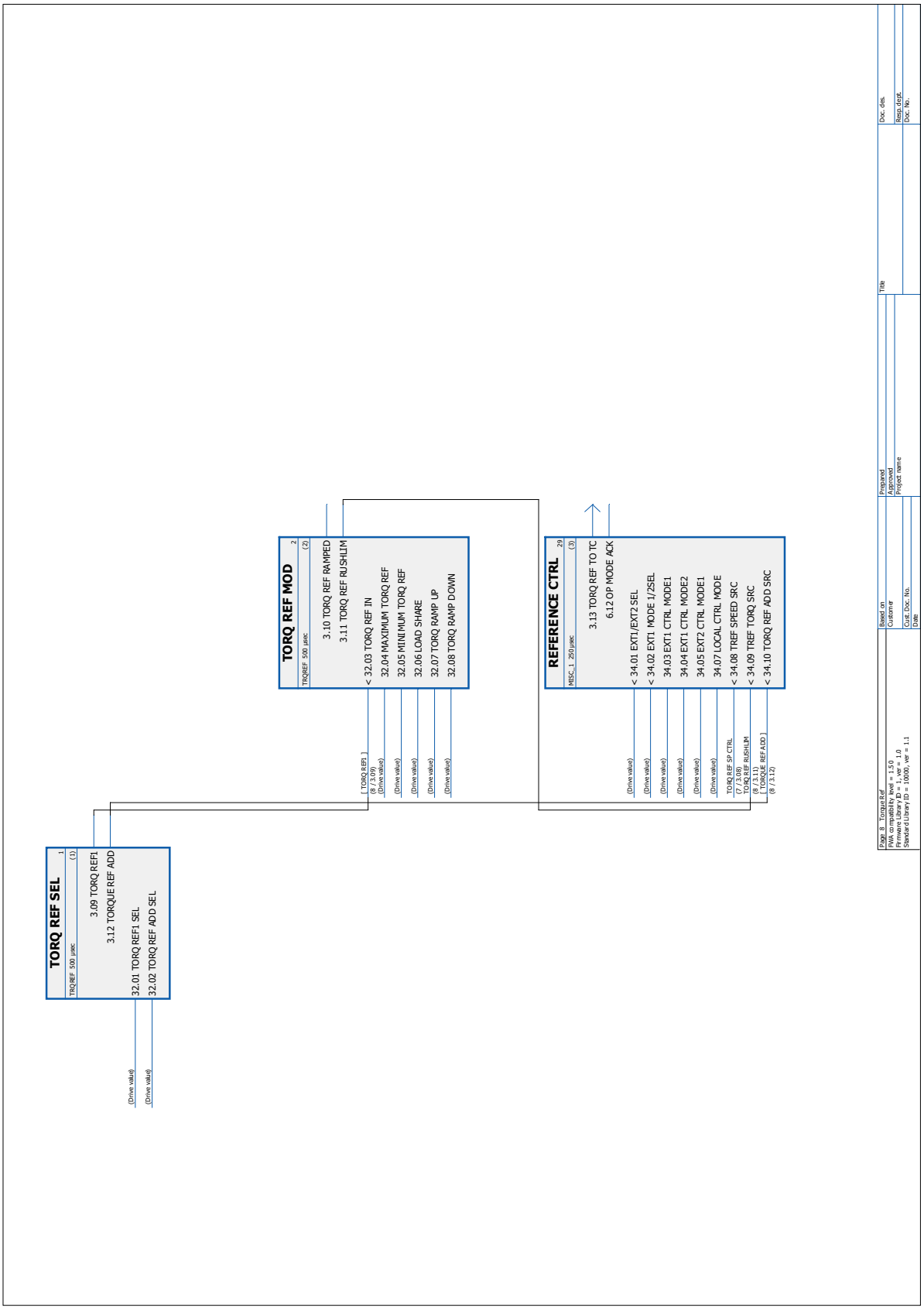

AO1
25
(8)
2.08 AO1
< 15.01 AO1 PTR
15.02 AO1 FILT TIME
15.03 AO1 MAX
15.04 AO1 MIN
15.05 AO1 MAX SCALE
15.06 AO1 MIN SCALE
[CURRENT PERC]
(-/ / 105)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
AO2
26
(9)
2.09 AO2
< 15.07 AO2 PTR
15.08 AO2 FILT TIME
15.09 AO2 MAX
15.10 AO2 MIN
15.11 AO2 MAX SCALE
15.12 AO2 MIN SCALE
[SPEED ACT PERC]
(-/ / 105)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
AI1
23
(6)
2.04 AI1
2.05 AI1 SCALED
13.01 AI1 FILT TIME
13.02 AI1 MAX
13.03 AI1 MIN
13.04 AI1 MAX SCALE
13.05 AI1 MIN SCALE
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
AI2
24
(7)
2.06 AI2
2.07 AI2 SCALED
13.06 AI2 FILT TIME
13.07 AI2 MAX
13.08 AI2 MIN
13.09 AI2 MAX SCALE
13.10 AI2 MIN SCALE
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
IOCTRL 2 msec
Page 3 Analog I/O
FWA compatibility level = 1.50
Firmware Library ID = 1, ver = 1.0
Standard Library ID = 10000, ver = 1.1
Based on
Customer
Cust. Doc. No.
Date
Prepared
Approved
Projectname
Title
Doc. des.
Resp. dept.
Doc. No.

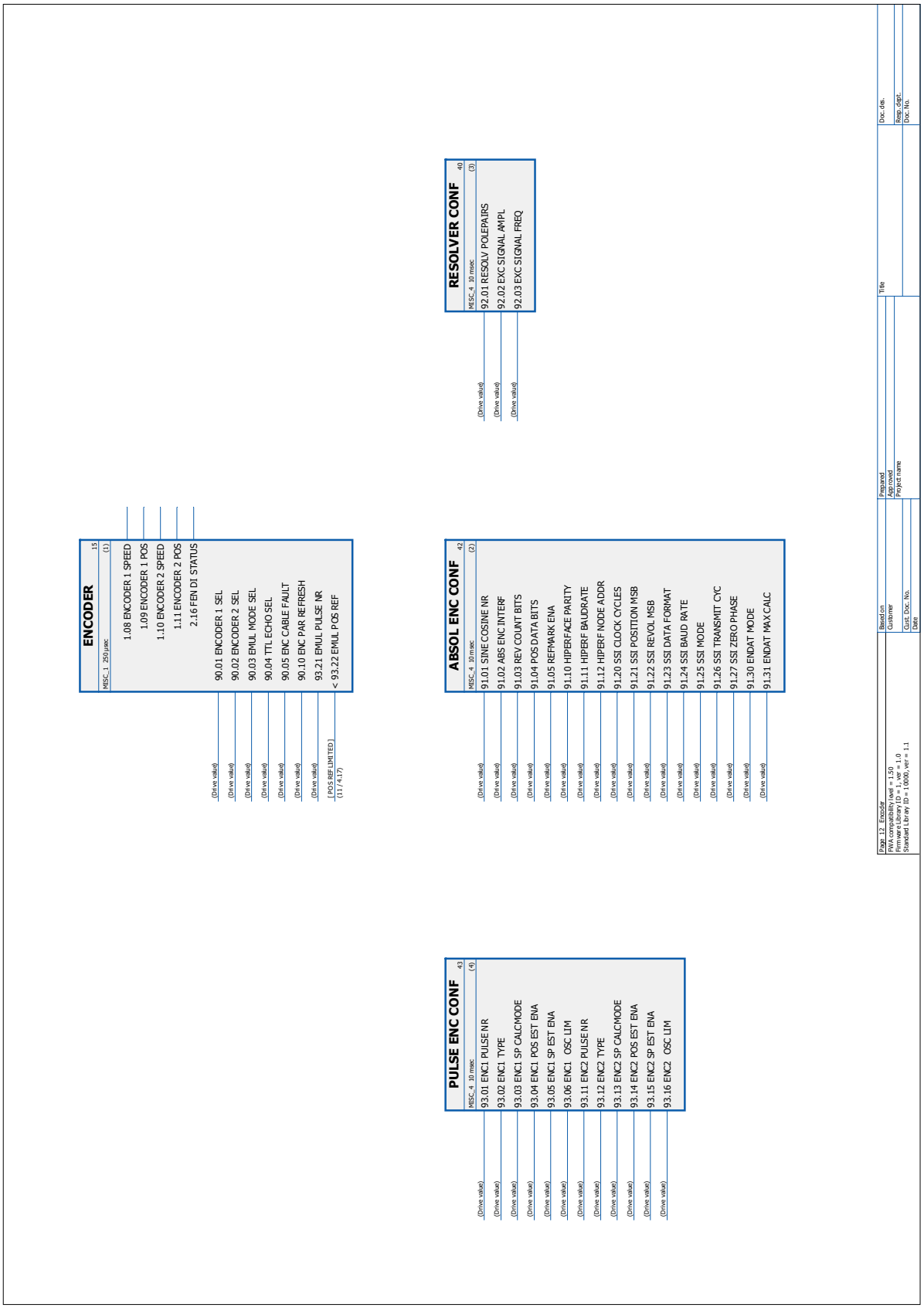

FIELDBUS
MISC_2 500 µsec
2.12 FBA MAIN CW
2.13 FBA MAIN SW
2.14 FBA MAIN REF1
2.15 FBA MAIN REF2
50.01 FBA ENABLE
50.02 COMM LOSS FUNC
50.03 COMM LOSS T OUT
50.04 FBA REF1 MODESEL
50.05 FBA REF2 MODESEL
50.06 FBA ACT1 TR SRC
50.07 FBA ACT2 TR SRC
50.08 FBA SW B12 SRC
50.09 FBA SW B13 SRC
50.10 FBA SW B14 SRC
50.11 FBA SW B15 SRC
DRIVE LOGIC
MISC_3 2 msec
2.18 D2D FOLLOWER CW
6.01 STATUS WORD 1
6.02 STATUS WORD 2
6.03 SPEED CTRL STAT
6.05 LIMIT WORD 1
6.07 TORQ LIM STATUS
6.09 POS CTRL STATUS
6.10 POS CTRL STATUS2
6.11 POS COMR STATUS
10.01 EXT1 START FUNC
10.02 EXT1 START IN1
10.03 EXT1 START IN2
10.04 EXT2 START FUNC
10.05 EXT2 START IN1
10.06 EXT2 START IN2
10.07 JOG1 START
10.08 FAULT RESET SEL
10.09 RUN ENABLE
10.10 EM STOP OFF3
10.11 EM STOP OFF1
10.12 START INHIBIT
10.13 FB CW USED
10.14 JOG2 START
10.15 JOG ENABLE
10.16 D2D CW USED
10.17 START ENABLE
D2D COMMUNICATION
MISC_2 500 µsec
2.17 D2D MAIN CW
2.19 D2D REF1
2.20 D2D REF2
57.01 LINK MODE
57.02 COMM LOSS FUNC
57.03 NODE ADDRESS
57.04 FOLLOWER MASK 1
57.05 FOLLOWER MASK 2
57.06 REF 1 SRC
57.07 REF 2 SRC
57.08 FOLLOWER CW SRC
57.09 KERNEL SYNC MODE
57.10 KERNEL SYNC OFFS
57.11 REF1 MSG TYPE
57.12 REF1 MC GROUP
57.13 NEXT REF1 MC GRP
57.14 NR REF1 MC GRPS
START/STOP MODE
MISC_3 2 msec
11.01 START MODE
11.02 DC MAGN TIME
11.03 STOP MODE
11.04 DC HOLD SPEED
11.05 DC HOLD CUR REF
11.06 DC HOLD
11.07 AUTOPHASING MODE
MECH BRAKE CTRL
MISC_3 2 msec
3.14 BRAKE TORQ MEM
3.15 BRAKE COMMAND
35.01 BRAKE CONTROL
35.02 BRAKE ACKNOWL
35.03 BRAKE OPEN DELAY
35.04 BRAKE CLOSE DLY
35.05 BRAKE CLOSE SPD
35.06 BRAKE OPEN TORQ
35.07 BRAKE CLOSE REQ
35.08 BRAKE OPEN HOLD
35.09 BRAKE FAULT FUNC
Page 4 Drive Logic
FWA compatibility level = 1.10
Firmware library ID = 1, ver = 1.0
Standard Library ID = 10000, ver = 1.1
Based on
Customer
Cust. Doc. No.
Date
Prepared
Approved
Project name
Title
Doc. des.
Res. dept.
Doc. No.

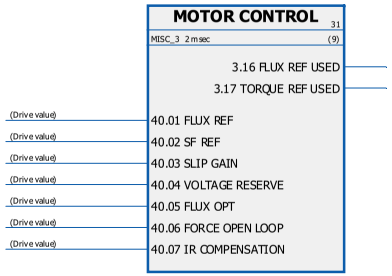

BRAKE CHOPPER
35
MISC_3 2 msec
(11)
48.01 BC ENABLE
< 48.02 BC RUN-TIME ENA
48.03 BRTHERMTIMECONST
48.04 BR POWER MAX CNT
48.05 R BR
48.06 BR TEMP FAULTLIM
48.07 BR TEMP ALARMLIM
(Drive value)
VOLTAGE CTRL
34
MISC_4 10 msec
(1)
1.19 USED SUPPLY VOLT
47.01 OVER/VOLTAGE CTRL
47.02 UNDERVOLT CTRL
47.03 SUPPLVOLTAUTO-ID
47.04 SUPPLY VOLTAGE
Page 5 Drive Control
FWA compatibility level = 1.50
Firmware Library ID = 1, ver = 1.0
Standard Library ID = 10000, ver = 1.1
Based on
Customer
Cust. Doc. No.
Date
Prepared
Approved
Projectname
Title
Doc. cls.
Resp. dept.
Doc. No.


SPEED REF SEL
3.01 SPEED REF1
3.02 SPEED REF2
24.01 SPEED REF1 SEL
24.02 SPEED REF2 SEL
SPEED REF MOD
3.03 SPEEDREF RAMP IN
24.03 SPEED REF1 IN
24.04 SPEED REF2 IN
24.05 SPEED REF 1/2SEL
24.06 SPEED SHARE
24.07 SPEEDREF NEG ENA
24.08 CONST SPEED
24.09 CONST SPEED ENA
24.10 SPEED REF JOG1
24.11 SPEED REF JOG2
24.12 SPEED REFMIN ABS
SPEED REF RAMP
3.04 SPEEDREF RAMPED
25.01 SPEED RAMP IN
25.02 SPEED SCALING
25.03 ACC TIME
25.04 DEC TIME
25.05 SHAPE TIME ACC1
25.06 SHAPE TIME ACC2
25.07 SHAPE TIME DEC1
25.08 SHAPE TIME DEC2
25.09 ACC TIME JOGGING
25.10 DEC TIME JOGGING
25.11 EM STOP TIME
25.12 SPEEDREF BAL
25.13 SPEEDREF BAL ENA
Page 6 Speed Ref
FWA compatibility level = 3.0
Firmware ID = 11000, ver = 1.0
Standard Library ID = 10000, ver = 1.1
Based on
Customer
Cust. Doc. No.
Date
Prepared
Approved
Project name
Title
Doc. des.
Resp. dept.
Doc. No.

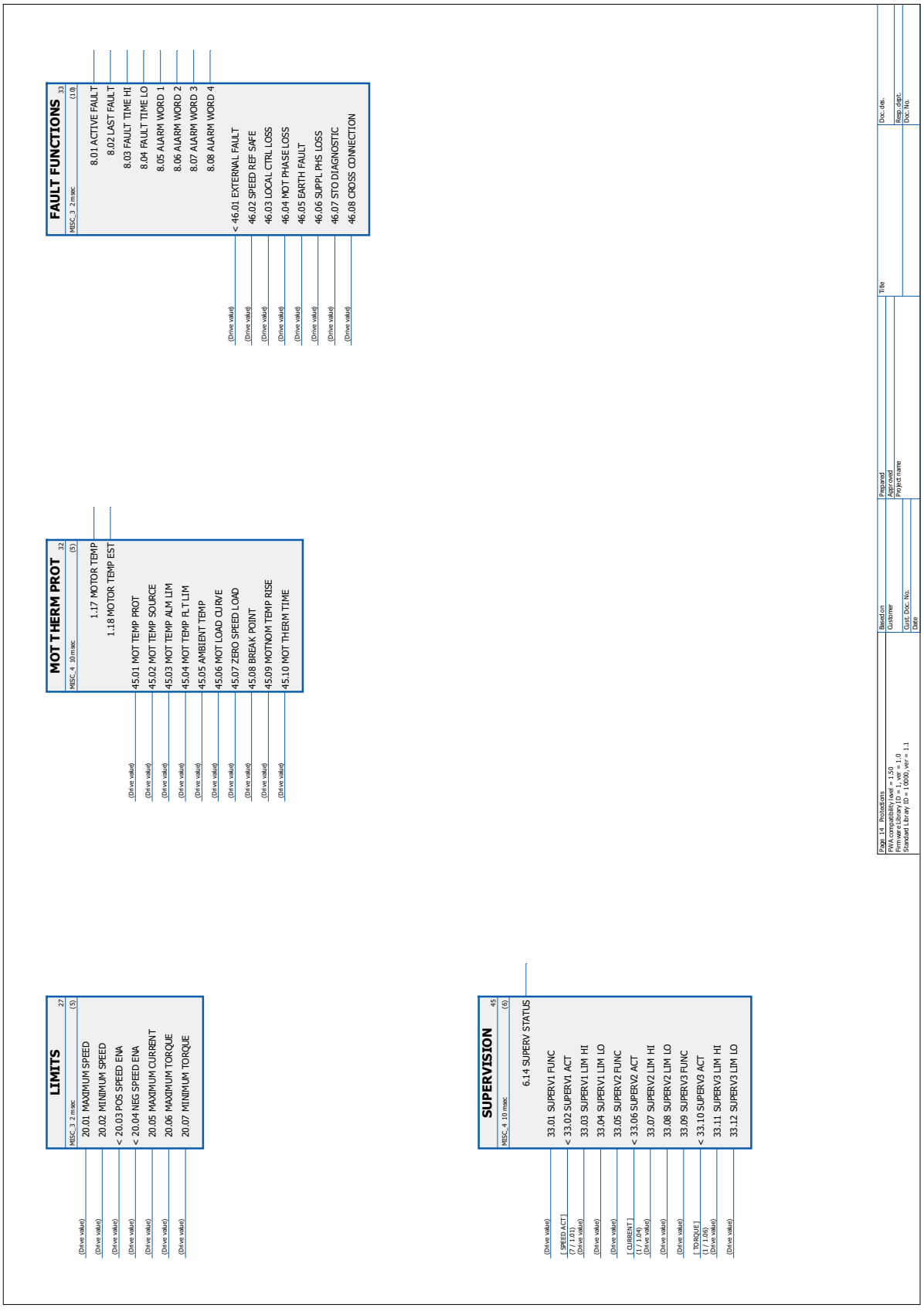

SPEED FEEDBACK
MISC_1 250 µsec
1.01 SPEED ACT
22.01 SPEED FB SEL
22.02 SPEED ACT FTIME
22.03 MOTOR GEAR MUL
22.04 MOTOR GEAR DIV
22.05 ZERO SPEED LIMIT
22.06 ZERO SPEED DELAY
22.07 ABOVE SPEED LIM
22.08 SPEED TRIPMARGIN
22.09 SPEED FB FAULT
SPEED ERROR
SPDCTR 250 µsec
3.05 SPEEDREF USED
3.06 SPEED ERROR FILT
3.07 ACC COMP TORQ
SPEED CONTROL
SPDCTR 250 µsec
3.08 TORQ REF SP CTRL
28.01 SPEED ERR NCTRL
28.02 PROPORT GAIN
28.03 INTEGRATION TIME
28.04 DERIVATION TIME
28.05 DERIV FILT TIME
28.06 ACC COMPENSATION
28.07 DROOPING RATE
28.08 BAL REFERENCE
28.09 SPEEDCTRL BAL EN
28.10 MIN TORQ SP CTRL
28.11 MAX TORQ SP CTRL
28.12 PI ADAPT MAX SPD
28.13 PI ADAPT MIN SPD
28.14 P GAIN ADPT COEF
28.15 I TIME ADPT COEF
SPEED ACT
SPEEDREF RAMPED
SPEED REF POS
SPEED FEED FWD
SPEED ERROR FILT
ACC COMP TORQ
(Drive value)
Prepared
Approved
Project name
Title
Doc. des.
Resp. dept.
Doc. No.
Based on
Customer
Cust. Doc. No.
Date
Page 7 Speed Ctrl
FWA compatibility level = 1.5.0
Firmware Library ID = 1, ver = 1.0
Standard Library ID = 10000, ver = 1.1


TORQ REF SEL
1
TRQREF 500 µsec
(1)
3.09 TORQ REF1
3.12 TORQUE REF ADD
32.01 TORQ REF1 SEL
32.02 TORQ REF ADD SEL
(Drive value)
(Drive value)
TORQ REF MOD
2
TRQREF 500 µsec
(2)
3.10 TORQ REF RAMPED
3.11 TORQ REF RUSHLIM
< 32.03 TORQ REF IN
32.04 MAXIMUM TORQ REF
32.05 MINIMUM TORQ REF
32.06 LOAD SHARE
32.07 TORQ RAMP UP
32.08 TORQ RAMP DOWN
[TORQ REF1]
(8 / 3.09)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
REFERENCE CTRL
29
MISC_1 250 µsec
(3)
3.13 TORQ REF TO TC
6.12 OP MODE ACK
< 34.01 EXT1/EXT2 SEL
< 34.02 EXT1 MODE 1/2SEL
34.03 EXT1 CTRL MODE1
34.04 EXT1 CTRL MODE2
34.05 EXT2 CTRL MODE1
34.07 LOCAL CTRL MODE
< 34.08 TREF SPEED SRC
< 34.09 TREF TORQ SRC
< 34.10 TORQ REF ADD SRC
(Drive value)
(Drive value)
(Drive value)
(Drive value)
(Drive value)
TORQ REF SP CTRL
(7 / 3.09)
TORQ REF RUSHLM
(8 / 3.11)
[TORQUE REF ADD]
(8 / 3.12)
Page 8 Torque Ref
FWA compatibility level = 1.50
Firmware name ID = FWA, ver = 1.0
Standard Library ID = 10000, ver = 1.1
Based on
Customer
Cust. Doc. No.
Date
Prepared
Approved
Project name
Title
Doc. des.
Resp. dept.
Doc. No.


ENCODER
MISC_1 250 µsec
(1)
1.08 ENCODER 1 SPEED
1.09 ENCODER 1 POS
1.10 ENCODER 2 SPEED
1.11 ENCODER 2 POS
2.16 FEN DI STATUS
90.01 ENCODER 1 SEL
90.02 ENCODER 2 SEL
90.03 EMUL MODE SEL
90.04 TTL ECHO SEL
90.05 ENC CABLE FAULT
90.10 ENC PAR REFRESH
93.21 EMUL PULSE NR
< 93.22 EMUL POS REF
(Drive value)
[POS REF LIMITED]
(1 / 4 L7)
RESOLVER CONF
MISC_4 10 msec
(3)
92.01 RESOLV POLEPAIRS
92.02 EXC SIGNAL AMPL
92.03 EXC SIGNAL FREQ
ABSOL ENC CONF
MISC_4 10 msec
(2)
91.01 SINE COSINE NR
91.02 ABS ENC INTERF
91.03 REV COUNT BITS
91.04 POS DATA BITS
91.05 REFMARK ENA
91.10 HIPERFACE PARITY
91.11 HIPERF BAUDRATE
91.12 HIPERF NODE ADDR
91.20 SSI CLOCK CYCLES
91.21 SSI POSITION MSB
91.22 SSI REVOL MSB
91.23 SSI DATA FORMAT
91.24 SSI BAUD RATE
91.25 SSI MODE
91.26 SSI TRANSMIT CYC
91.27 SSI ZERO PHASE
91.30 ENDAT MODE
91.31 ENDAT MAX CALC
PULSE ENC CONF
MISC_4 10 msec
(4)
93.01 ENC1 PULSE NR
93.02 ENC1 TYPE
93.03 ENC1 SP CALCMODE
93.04 ENC1 POS EST ENA
93.05 ENC1 SP EST ENA
93.06 ENC1 OSC LIM
93.11 ENC2 PULSE NR
93.12 ENC2 TYPE
93.13 ENC2 SP CALCMODE
93.14 ENC2 POS EST ENA
93.15 ENC2 SP EST ENA
93.16 ENC2 OSC LIM
Page 12 Encoder
FWA compatibility (last = 1.50
Firmware Library ID = 1, ver = 1.0
Standard Library ID = 1 0000, ver = 1.1
Beadon
Customer
Cust. Doc. No.
Date
Prepared
Approved
Project name
Title
Doc. des.
Resp. dept.
Doc. No.



Page 13 MotorCtrl	Based on	Prepared	Title	Doc. des.
FWA compatibility level = 1.50	Customer	Approved		
Firmware Library ID = 1, ver = 1.0		Project name		Resp. dept.
Standard Library ID = 10000, ver = 1.1	Cust. Doc. No.			Doc. No.
	Date			


FAULT FUNCTIONS
MISC_3 2 msec
8.01 ACTIVE FAULT
8.02 LAST FAULT
8.03 FAULT TIME HI
8.04 FAULT TIME LO
8.05 ALARM WORD 1
8.06 ALARM WORD 2
8.07 ALARM WORD 3
8.08 ALARM WORD 4
< 46.01 EXTERNAL FAULT
46.02 SPEED REF SAFE
46.03 LOCAL CTRL LOSS
46.04 MOT PHASE LOSS
46.05 EARTH FAULT
46.06 SUPPL PHS LOSS
46.07 STO DIAGNOSTIC
46.08 CROSS CONNECTION
MOT THERM PROT
MISC_4 10 msec
1.17 MOTOR TEMP
1.18 MOTOR TEMP EST
45.01 MOT TEMP PROT
45.02 MOT TEMP SOURCE
45.03 MOT TEMP ALM LIM
45.04 MOT TEMP FLT LIM
45.05 AMBIENT TEMP
45.06 MOT LOAD CURVE
45.07 ZERO SPEED LOAD
45.08 BREAK POINT
45.09 MOTNOM TEMP RISE
45.10 MOT THERM TIME
LIMITS
MISC_3 2 msec
20.01 MAXIMUM SPEED
20.02 MINIMUM SPEED
< 20.03 POS SPEED ENA
< 20.04 NEG SPEED ENA
20.05 MAXIMUM CURRENT
20.06 MAXIMUM TORQUE
20.07 MINIMUM TORQUE
SUPERVISION
MISC_4 10 msec
6.14 SUPERV STATUS
33.01 SUPERV1 FUNC
< 33.02 SUPERV1 ACT
33.03 SUPERV1 LIM HI
33.04 SUPERV1 LIM LO
33.05 SUPERV2 FUNC
< 33.06 SUPERV2 ACT
33.07 SUPERV2 LIM HI
33.08 SUPERV2 LIM LO
33.09 SUPERV3 FUNC
< 33.10 SUPERV3 ACT
33.11 SUPERV3 LIM HI
33.12 SUPERV3 LIM LO
Page 14 Protection
FW& compatibility (sw) = 1.50
Firmware Library ID = 1, ver = 1.0
Standard Library ID = 10000, ver = 1.1
Reason
Customer
Cust. Doc. No.
Date
Prepared
Approved
Project name
Title
Doc. des.
Resp. dept.
Doc. No.

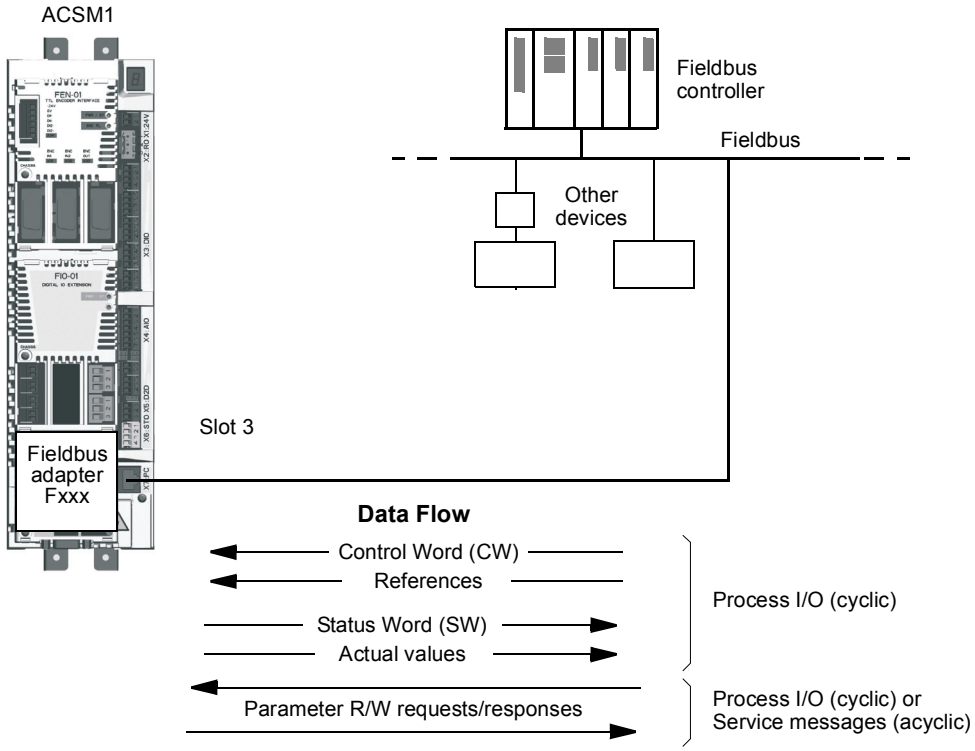
Appendix A – Fieldbus control

What this chapter contains

The chapter describes how the drive can be controlled by external devices over a communication network.

System overview

The drive can be connected to a fieldbus controller via a fieldbus adapter module. The adapter module is connected to drive Slot 3.



The drive can be set to receive all of its control information through the fieldbus interface, or the control can be distributed between the fieldbus interface and other available sources, for example digital and analogue inputs.

The drive can communicate with fieldbus controller via fieldbus adapter using one of the following serial communication protocols:

- PROFIBUS-DP® (FPBA-01 adapter)
- CANopen® (FCAN-01 adapter)
- DeviceNet® (FDNA-01 adapter)
- Modbus/RTU (FSCA-01 adapter)
- Modbus/TCP (FENA-01 adapter).

Setting up communication through a fieldbus adapter module

Before configuring the drive for fieldbus control, the adapter module must be mechanically and electrically installed according to the instructions given in the User's Manual of the appropriate fieldbus adapter module.

The communication between the drive and the fieldbus adapter module is activated by setting parameter 50.01 FBA ENABLE to (1) Enable. The adapter-specific parameters must also be set. See the table below.

Parameter	Setting for fieldbus control	Function/Information
COMMUNICATION INITIALISATION AND SUPERVISION		
50.01 FBA ENABLE	(1) Enable	Initialises communication between drive and fieldbus adapter module.
50.02 COMM LOSS FUNC	(0) No (1) Fault (2) Spd ref Safe (3) Last speed	Selects how the drive reacts in a fieldbus communication break.
50.03 COMM LOSS T OUT	0.3…6553.5 s	Defines the time between communication break detection and the action selected with parameter 50.02 COMM LOSS FUNC.
50.04 FBA REF1 MODESEL and 50.05 FBA REF2 MODESEL	(0) Raw data (1) Torque (2) Speed (5) Auto	Defines the fieldbus reference scaling. When (0) Raw data is selected, see also parameters 50.06…50.11. When both parameters are set to (5) Auto, the scalings for fieldbus references are set automatically according to parameter 34.03 EXT1 CTRL MODE1 as follows: FBA REF1 = Speed, FBA REF2 = Torque
ADAPTER MODULE CONFIGURATION		
51.01 FBA TYPE	–	Displays the type of the fieldbus adapter module.
51.02 FBA PAR2 • • • 51.26 FBA PAR26	These parameters are adapter module-specific. For more information, see the User's Manual of the fieldbus adapter module. Note that not all of these parameters are necessarily used.	
51.27 FBA PAR REFRESH	(0) DONE (1) REFRESH	Validates any changed adapter module configuration parameter settings.
51.28 PAR TABLE VER	–	Displays the parameter table revision of the fieldbus adapter module mapping file stored in the memory of the drive.
51.29 DRIVE TYPE CODE	–	Displays the drive type code of the fieldbus adapter module mapping file stored in the memory of the drive.
51.30 MAPPING FILE VER	–	Displays the fieldbus adapter module mapping file revision stored in the memory of the drive.
51.31 D2FBA COMM STA	–	Displays the status of the fieldbus adapter module communication.
51.32 FBA COMM SW VER	–	Displays the common program revision of the adapter module.
51.33 FBA APPL SW VER	–	Displays the application program revision of the adapter module.
Note: In the User's Manual of the fieldbus adapter module, the parameter group number is 1 or A for parameters 51.01…51.26.		

Parameter	Setting for fieldbus control	Function/Information
TRANSMITTED DATA SELECTION		
52.01 FBA DATA IN1 … 52.12 FBA DATA IN12	0 4…6 14…16 101…9999	Defines the data transmitted from drive to fieldbus controller. **Note:** If the selected data is 32 bits long, two parameters are reserved for the transmission.
53.01 FBA DATA OUT1 … 53.12 FBA DATA OUT12	0 1…3 11…13 1001…9999	Defines the data transmitted from fieldbus controller to drive. **Note:** If the selected data is 32 bits long, two parameters are reserved for the transmission.
Note: In the User's Manual of the fieldbus adapter module, the parameter group number is 2 or B for parameters 52.01…52.12 and 3 or C for parameters 53.01…53.12.		

After the module configuration parameters have been set, the drive control parameters (see section Drive control parameters) must be checked and adjusted when necessary.

The new settings will take effect when the drive is powered up the next time (before powering off the drive, wait at least 1 minute), or when parameter 51.27 FBA PAR REFRESH is activated.

Drive control parameters

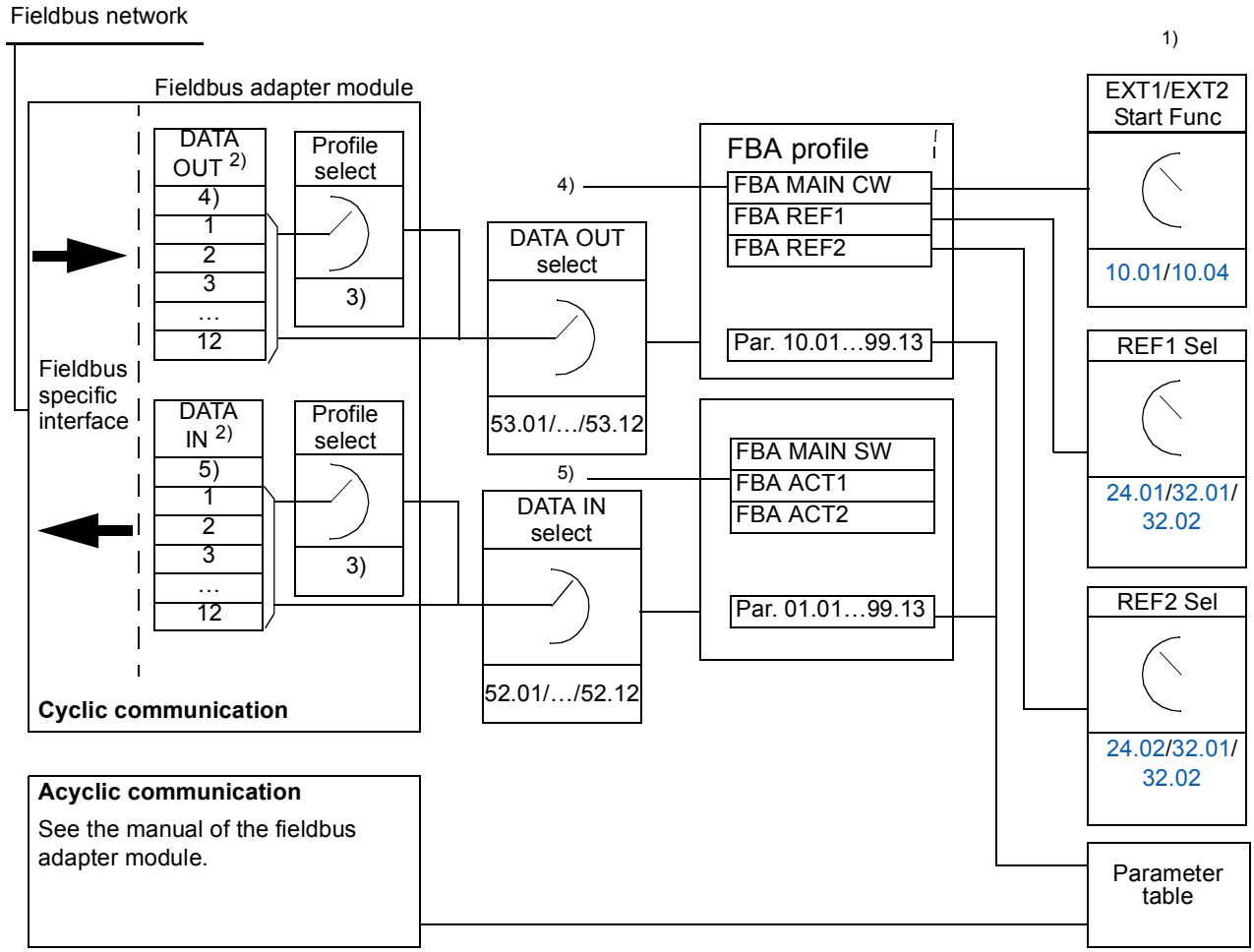
The Setting for fieldbus control column gives the value to use when the fieldbus interface is the desired source or destination for that particular signal. The Function/Information column gives a description of the parameter.

Parameter	Setting for fieldbus control	Function/Information
CONTROL COMMAND SOURCE SELECTION		
10.01 EXT1 START FUNC	(3) FBA	Selects fieldbus as the source for the start and stop commands when EXT1 is selected as the active control location.
10.04 EXT2 START FUNC	(3) FBA	Selects fieldbus as the source for the start and stop commands when EXT2 is selected as the active control location.
24.01 SPEED REF1 SEL	(3) FBA REF1 (4) FBA REF2	Fieldbus reference REF1 or REF2 is used as speed reference 1.
24.02 SPEED REF2 SEL	(3) FBA REF1 (4) FBA REF2	Fieldbus reference REF1 or REF2 is used as speed reference 2.
32.01 TORQ REF1 SEL	(3) FBA REF1 (4) FBA REF2	Fieldbus reference REF1 or REF2 is used as torque reference 1.
32.02 TORQ REF ADD SEL	(3) FBA REF1 (4) FBA REF2	Fieldbus reference REF1 or REF2 is used for torque reference addition.
SYSTEM CONTROL INPUTS		
16.07 PARAM SAVE	(0) Done (1) Save	Saves parameter value changes (including those made through fieldbus control) to permanent memory.

The fieldbus control interface

The cyclic communication between a fieldbus system and the drive consists of 16/32-bit input and output data words. The drive supports at the maximum the use of 12 data words (16-bit) in each direction.

Data transmitted from the drive to the fieldbus controller is defined by parameters 52.01…52.12 (FBA DATA IN) and data transmitted from the fieldbus controller to the drive is defined by parameters 53.01…53.12 (FBA DATA OUT).



1) See also other parameters which can be controlled by the fieldbus.
2) The maximum number of used data words is protocol-dependent.
3) Profile/instance selection parameters. Fieldbus module specific parameters. For more information, see the User's Manual of the appropriate fieldbus adapter module.
4) With DeviceNet the control part is transmitted directly.
5) With DeviceNet the actual value part is transmitted directly.

The Control Word and the Status Word

The Control Word (CW) is the principal means of controlling the drive from a fieldbus system. The Control Word is sent by the fieldbus controller to the drive. The drive switches between its states according to the bit-coded instructions of the Control Word.

The Status Word (SW) is a word containing status information, sent by the drive to the fieldbus controller.

Actual values

Actual values (ACT) are 16/32-bit words containing information on selected operations of the drive.

FBA communication profile

The FBA communication profile is a state machine model which describes the general states and state transitions of the drive. The State diagram on page 330 presents the most important states (including the FBA profile state names). The FBA Control Word (2.12 FBA MAIN CW, page 63) commands the transitions between these states and the FBA Status Word (2.13 FBA MAIN SW, page 65) indicates the status of the drive.

Fieldbus adapter module profile (selected by adapter module parameter) defines how the control word and status word are transmitted in a system which consists of fieldbus controller, fieldbus adapter module and drive. With transparent modes, control word and status word are transmitted without any conversion between the fieldbus controller and the drive. With other profiles (e.g. PROFIdrive for FPBA-01, AC/DC drive for FDNA-01, DS-402 for FCAN-01 and ABB Drives profile for all fieldbus adapter modules) fieldbus adapter module converts the fieldbus-specific control word to the FBA communication profile and status word from FBA communication profile to the fieldbus-specific status word.

For descriptions of other profiles, see the *User's Manual* of the appropriate fieldbus adapter module.

Fieldbus references

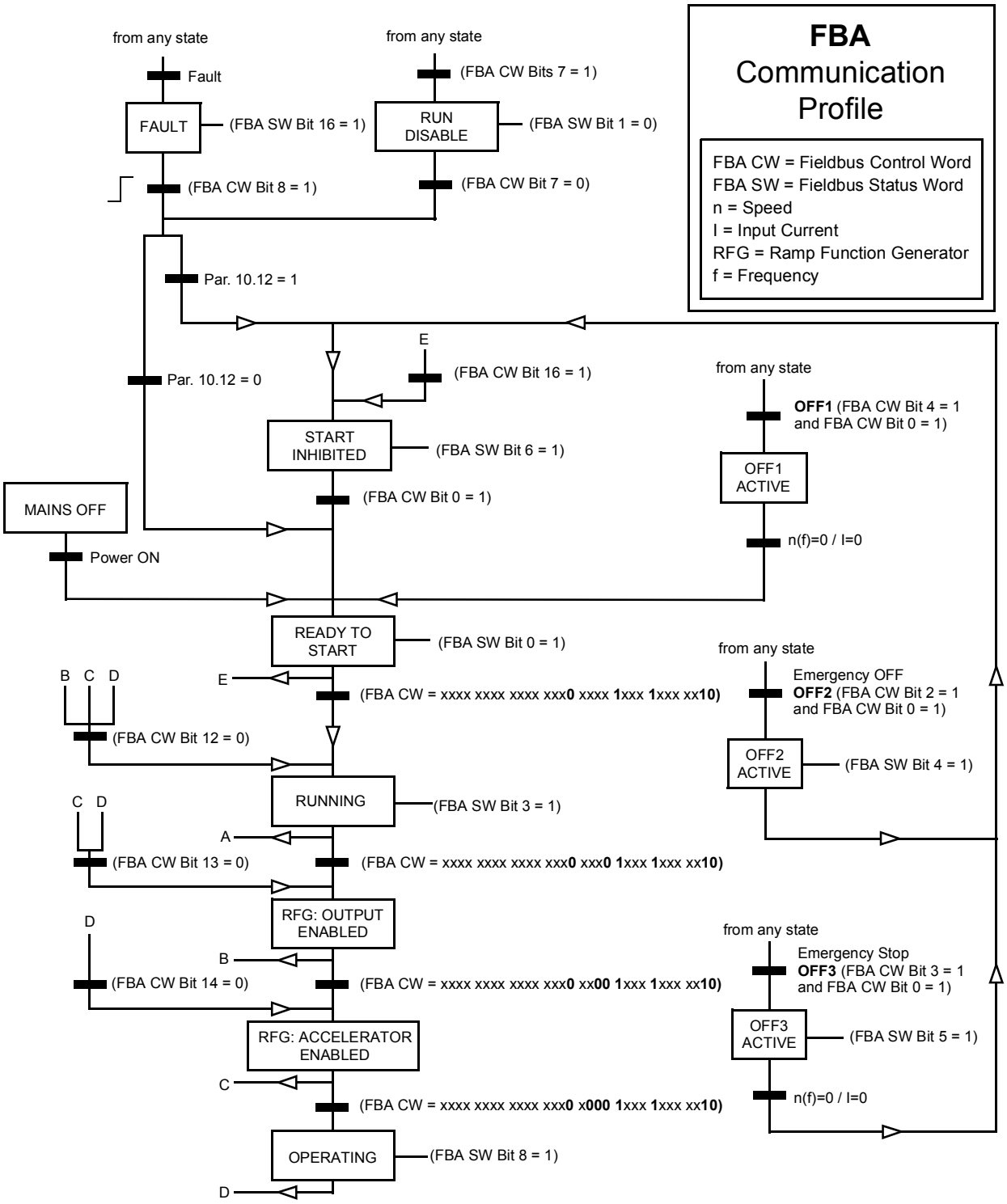
References (FBA REF) are 16/32-bit signed integers. A negative reference is formed by calculating the two's complement from the corresponding positive reference value. The contents of each reference word can be used as speed or torque reference.

When torque or speed reference scaling is selected (by parameter 50.04 FBA REF1 MODESEL / 50.05 FBA REF2 MODESEL), the fieldbus references are 32-bit integers. The value consists of a 16-bit integer value and a 16-bit fractional value. The speed/torque reference scaling is as follows:

Reference	Scaling	Notes
Torque reference	FBA REF / 65536 (value in %)	Final reference is limited by parameters 20.06 MAXIMUM TORQUE and 20.07 MINIMUM TORQUE.
Speed reference	FBA REF / 65536 (value in rpm)	Final reference is limited by parameters 20.01 MAXIMUM SPEED, 20.02 MINIMUM SPEED and 24.12 SPEED REFMIN ABS.

State diagram

The following presents the state diagram for the FBA communication profile. For other profiles, see the *User's Manual* of the appropriate fieldbus adapter module.


FBA
Communication
Profile
FBA CW = Fieldbus Control Word
FBA SW = Fieldbus Status Word
n = Speed
I = Input Current
RFG = Ramp Function Generator
f = Frequency
from any state
Fault
FAULT
(FBA SW Bit 16 = 1)
(FBA CW Bit 8 = 1)
from any state
(FBA CW Bits 7 = 1)
RUN DISABLE
(FBA SW Bit 1 = 0)
(FBA CW Bit 7 = 0)
Par. 10.12 = 1
Par. 10.12 = 0
E
(FBA CW Bit 16 = 1)
START INHIBITED
(FBA SW Bit 6 = 1)
(FBA CW Bit 0 = 1)
MAINS OFF
Power ON
from any state
OFF1 (FBA CW Bit 4 = 1 and FBA CW Bit 0 = 1)
OFF1 ACTIVE
n(f)=0 / I=0
READY TO START
(FBA SW Bit 0 = 1)
B C D
E
(FBA CW = xxxx xxxx xxxx xxx0 xxxx 1xxx 1xxx xx10)
(FBA CW Bit 12 = 0)
from any state
Emergency OFF
OFF2 (FBA CW Bit 2 = 1 and FBA CW Bit 0 = 1)
OFF2 ACTIVE
(FBA SW Bit 4 = 1)
RUNNING
(FBA SW Bit 3 = 1)
C D
A
(FBA CW Bit 13 = 0)
(FBA CW = xxxx xxxx xxxx xxx0 xxx0 1xxx 1xxx xx10)
RFG: OUTPUT ENABLED
D
B
(FBA CW Bit 14 = 0)
(FBA CW = xxxx xxxx xxxx xxx0 xx00 1xxx 1xxx xx10)
from any state
Emergency Stop
OFF3 (FBA CW Bit 3 = 1 and FBA CW Bit 0 = 1)
OFF3 ACTIVE
(FBA SW Bit 5 = 1)
n(f)=0 / I=0
RFG: ACCELERATOR ENABLED
C
(FBA CW = xxxx xxxx xxxx xxx0 x000 1xxx 1xxx xx10)
OPERATING
(FBA SW Bit 8 = 1)
D

Appendix B – Drive-to-drive link

What this chapter contains

This chapter describes the wiring of, and available communication methods on the drive-to-drive link. Examples of using standard function blocks in the communication are also given starting on page 339.

General

The drive-to-drive link is a daisy-chained RS-485 transmission line, constructed by connecting the X5 terminal blocks of the JCU Control Units of several drives. It is also possible to use an FMBA Modbus extension module installed into an option slot on the JCU. The firmware supports up to 63 nodes on the link.

The link has one master drive; the rest of the drives are followers. By default, the master broadcasts control commands as well as speed and torque references for all followers. The master can send 8 messages per millisecond at 100/150-microsecond intervals. Sending one message takes approximately 15 microseconds, which results in a theoretical link capacity of roughly 6 messages per 100 microseconds.

Multicasting the control data and reference 1 to a pre-defined group of drives is possible, as is chained multicast messaging. Reference 2 is always broadcast by the master to all followers. See parameters 57.11…57.14.

Wiring

Shielded twisted-pair cable (~100 ohm, e.g. PROFIBUS-compatible cable) must be used for the wiring. The maximum length of the link is 50 metres (164 ft).

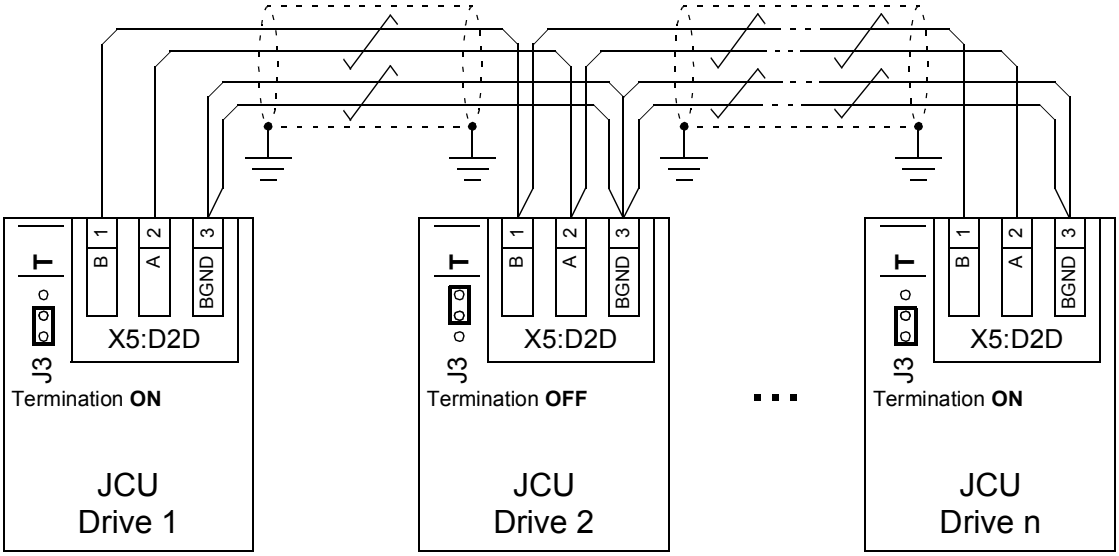
The JCU Control Unit has a jumper (J3, “T”) next to the X5 terminal block for bus termination. Termination must be ON on the drives at the ends of the drive-to-drive link; on intermediate drives, termination must be OFF.

Instead of the X5 connector, an FMBA Modbus extension module can be used.

For best immunity, high quality cable is recommended. The cable should be kept as short as possible. Unnecessary loops and running the cable near power cables (such as motor cables) must be avoided.

Note: The cable shields are to be grounded to the control cable clamp plate on the drive. Follow the instructions given in the *Hardware Manual* of the drive.

The following diagram shows the wiring of the drive-to-drive link.



Datasets

Drive-to-drive communication uses DDCS (Distributed Drives Communication System) messages and dataset tables for data transfer. Each drive has a dataset table of 256 datasets, numbered 0…255. Each dataset contains 48 data bits.

By default, datasets 0…15 and 200…255 are reserved for the drive firmware; datasets 16…199 are available for the user application program.

The contents of the two firmware communication datasets can be configured freely with pointer parameters and/or application programming with the DriveSPC tool. The 16-bit control word and 32-bit drive-to-drive reference 1 are transmitted from one dataset on a 500-microsecond (by default) time level; drive-to-drive reference 2 (32 bits) is transmitted from the other dataset on a 2-millisecond (by default) time level. Depending on the drive control mode, the followers can be configured to use the drive-to-drive commands and references with the following parameters:

Control data	Parameter	Setting for drive-to-drive communication
Start/Stop commands	10.01 EXT1 START FUNC 10.04 EXT2 START FUNC	(4) D2D
Speed reference	24.01 SPEED REF1 SEL 24.02 SPEED REF2 SEL	(5) D2D REF1 or (6) D2D REF2
Torque reference	32.01 TORQ REF1 SEL 32.02 TORQ REF ADD SEL	(5) D2D REF1 or (6) D2D REF2

The communication status of the followers can be supervised by a periodic supervision message from the master to the individual followers (see parameters 57.04 FOLLOWER MASK 1 and 57.05 FOLLOWER MASK 2).

Drive-to-drive function blocks can be used in the DriveSPC tool to enable additional communication methods (such as follower-to-follower messaging) and to modify the

use of datasets between the drives. See the function blocks under *Communication* (page 246).

Types of messaging

Each drive on the link has a unique node address allowing point-to-point communication between two drives. The node address 0 is automatically assigned to the master drive; on other drives, the node address is defined by parameter 57.03 NODE ADDRESS.

Multicast addressing is supported, allowing the composition of groups of drives. Data sent to a multicast address is received by all drives that have that address. A multicast group can consist of 1…62 drives.

In broadcast messaging, data can be sent to all drives (actually, all followers) on the link.

Both master-to-follower(s) and follower-to-follower(s) communication is supported. A follower can send one message to another follower (or a group of followers) after receiving a token message from the master.

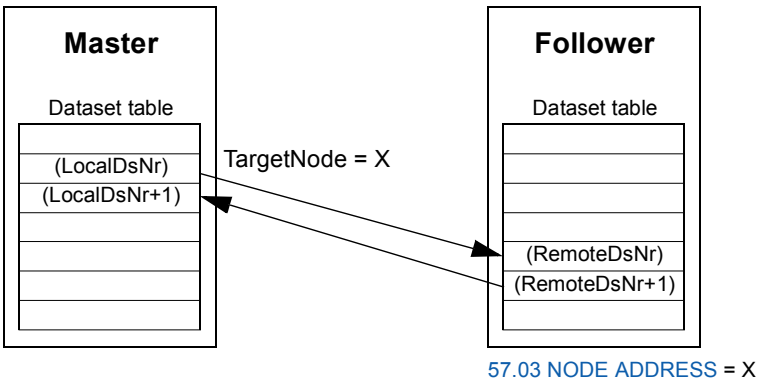
Type of messaging		**Note**
Point-to-point	Master point-to-point	Supported only at master
	Read remote	Supported only at master
	Follower point-to-point	Supported only at followers
Standard multicast		For both master and followers
Broadcast		For both master and followers
Token message for follower-to-follower communication		–
Chained multicast		Supported only for drive-to-drive reference 1 and control word

Master point-to-point messaging

In this type of messaging, the master sends one dataset (LocalDsNr) from its own dataset table to the follower's. TargetNode stands for the node address of the follower; RemoteDsNr specifies the target dataset number.

The follower responds by returning the contents of the next dataset. The response is stored into dataset LocalDsNr+1 in the master.

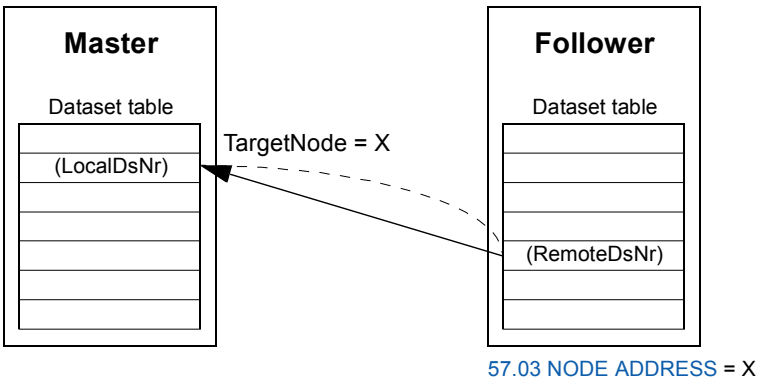
Note: Master point-to-point messaging is only supported at the master because the response is always sent to node address 0 (the master).



Read remote messaging

The master can read a dataset (RemoteDsNr) from a follower specified by TargetNode. The follower returns the contents of the requested dataset to the master. The response is stored at dataset LocalDsNr in the master.

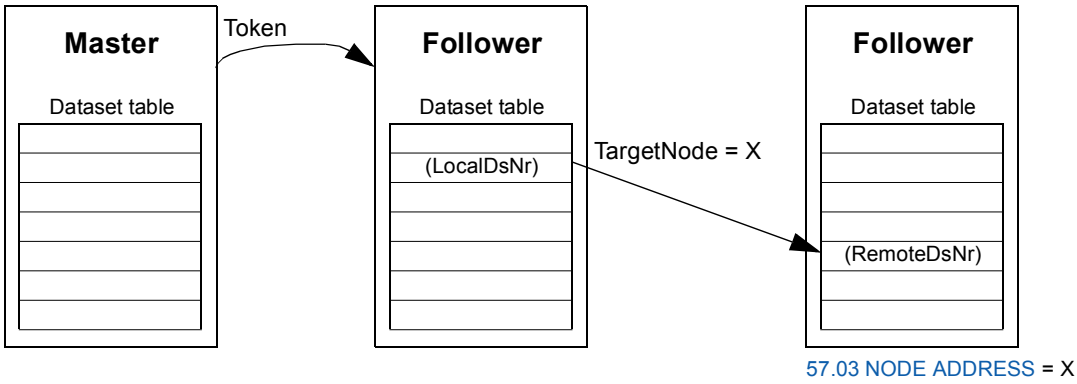
Note: Read remote messaging is only supported at the master because the response is always sent to node address 0 (the master).



Follower point-to-point messaging

This type of messaging is for point-to-point communication between followers. After receiving a token from the master, a follower can send one dataset to another follower with a follower point-to-point message. The target drive is specified using the node address.

Note: The data is not sent to the master.



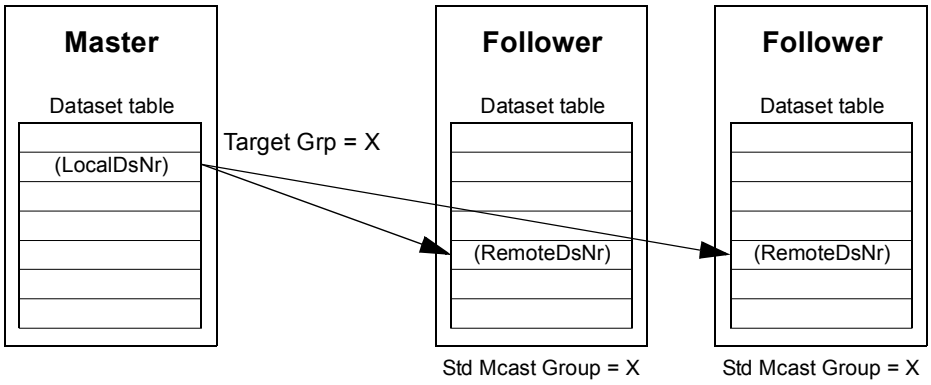
Standard multicast messaging

In standard multicast messaging, one dataset can be sent to a group of drives having the same standard multicast group address. The target group is defined by the D2D_Conf standard function block (see page 246).

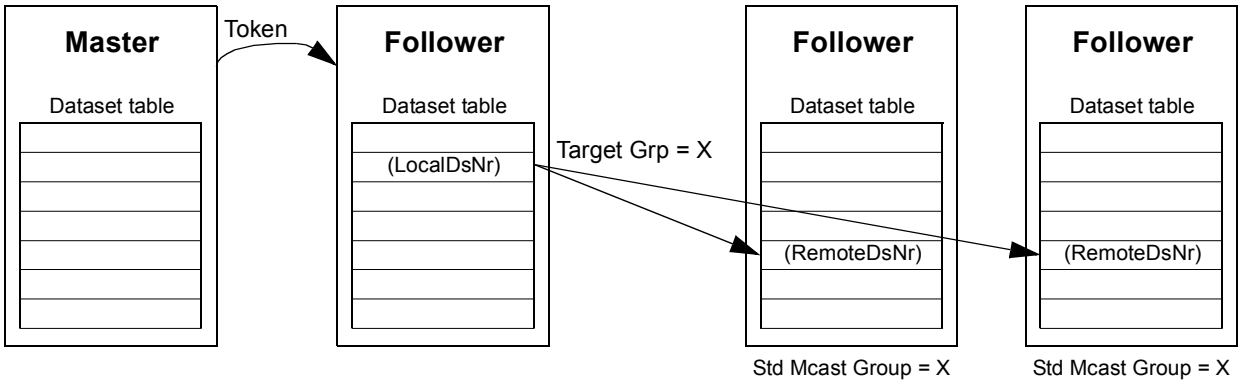
The sending drive can either be the master, or a follower after receiving a token from the master.

Note: The master does not receive the sent data even if it is a member of the target multicast group.

Master-to-follower(s) multicasting



Follower-to-follower(s) multicasting



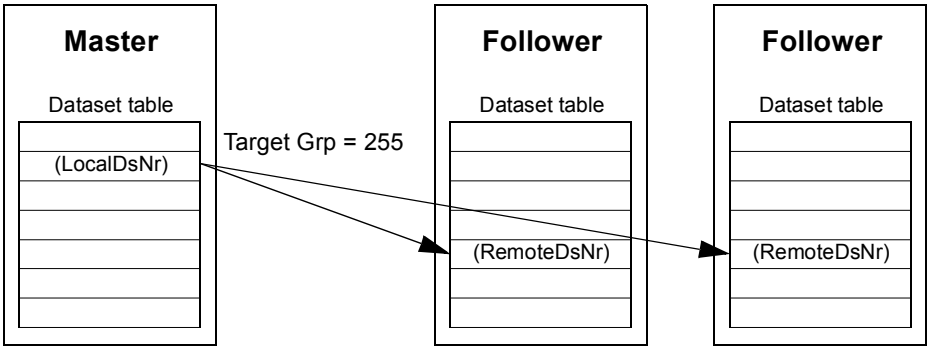
Broadcast messaging

In broadcasting, the master sends one dataset to all followers, or a follower sends one dataset to all other followers (after receiving a token from the master).

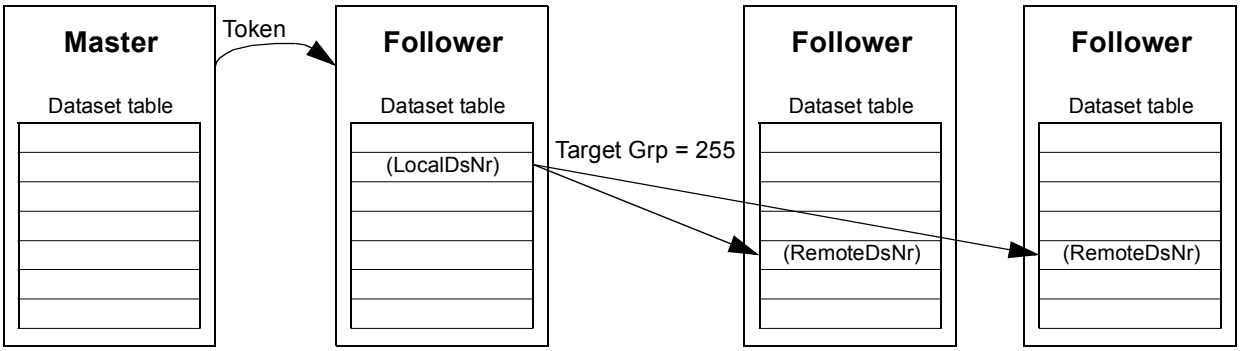
The target (Target Grp) is automatically set to 255 denoting all followers.

Note: The master does not receive any data broadcast by the followers.

Master-to-follower(s) broadcasting



Follower-to-follower(s) broadcasting



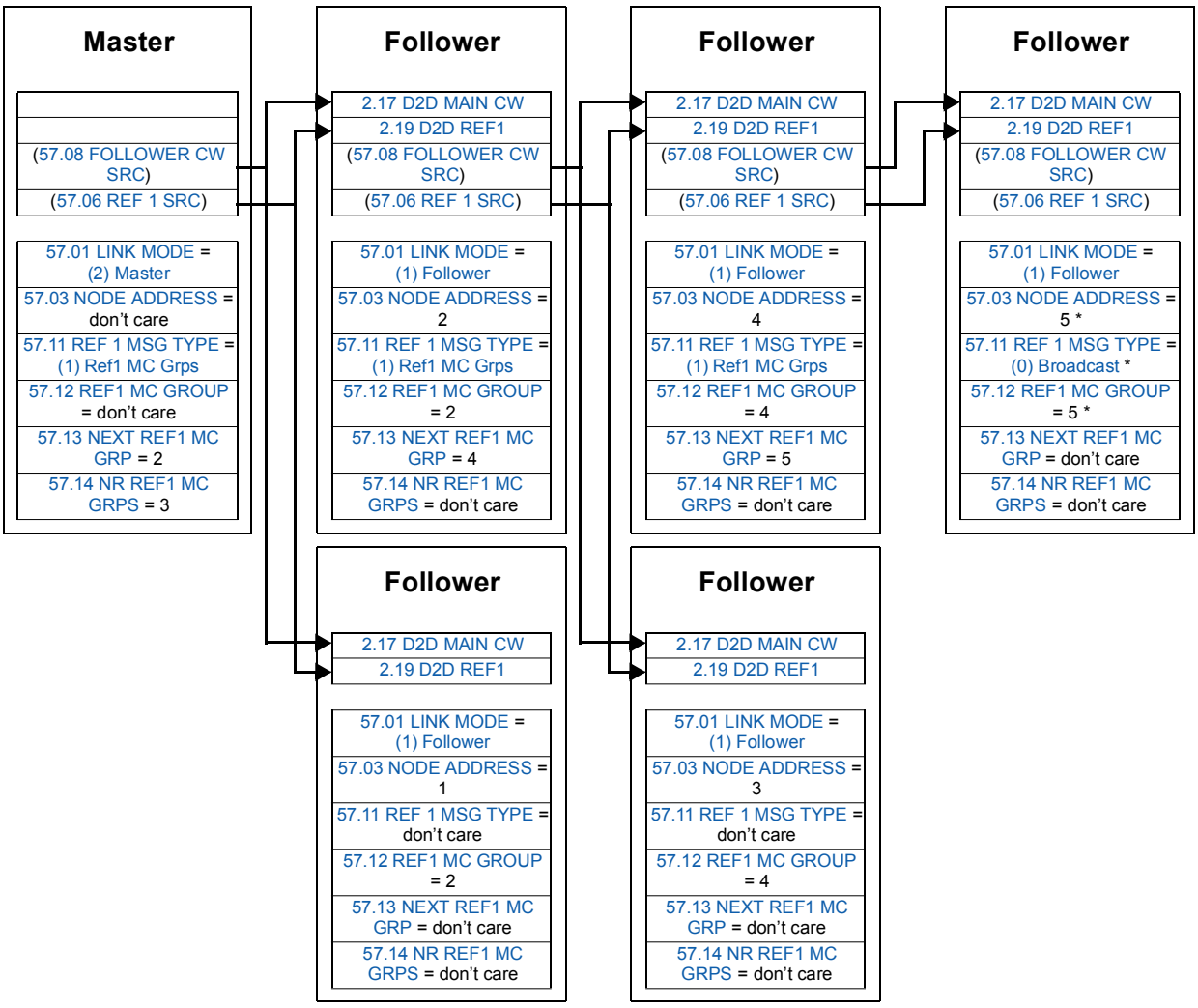
Chained multicast messaging

Chained multicasting is supported only for drive-to-drive reference 1 by the firmware.

The message chain is always started by the master. The target group is defined by parameter 57.13 NEXT REF1 MC GRP. The message is received by all followers that have parameter 57.12 REF1 MC GROUP set to the same value as parameter 57.13 NEXT REF1 MC GRP in the master.

If a follower have parameters 57.03 NODE ADDRESS and 57.12 REF1 MC GROUP set to the same value, it becomes a submaster. Immediately after a submaster receives the multicast message, it sends its own message to the next multicast group defined by parameter 57.13 NEXT REF1 MC GRP.

The duration of the entire message chain is approximately 15 microseconds multiplied by the number of links in the chain (defined by parameter 57.14 NR REF1 MC GRPS in the master).


Master
(57.08 FOLLOWER CW SRC)
(57.06 REF 1 SRC)
57.01 LINK MODE = (2) Master
57.03 NODE ADDRESS = don't care
57.11 REF 1 MSG TYPE = (1) Ref1 MC Grps
57.12 REF1 MC GROUP = don't care
57.13 NEXT REF1 MC GRP = 2
57.14 NR REF1 MC GRPS = 3
Follower
2.17 D2D MAIN CW
2.19 D2D REF1
(57.08 FOLLOWER CW SRC)
(57.06 REF 1 SRC)
57.01 LINK MODE = (1) Follower
57.03 NODE ADDRESS = 2
57.11 REF 1 MSG TYPE = (1) Ref1 MC Grps
57.12 REF1 MC GROUP = 2
57.13 NEXT REF1 MC GRP = 4
57.14 NR REF1 MC GRPS = don't care
Follower
2.17 D2D MAIN CW
2.19 D2D REF1
(57.08 FOLLOWER CW SRC)
(57.06 REF 1 SRC)
57.01 LINK MODE = (1) Follower
57.03 NODE ADDRESS = 4
57.11 REF 1 MSG TYPE = (1) Ref1 MC Grps
57.12 REF1 MC GROUP = 4
57.13 NEXT REF1 MC GRP = 5
57.14 NR REF1 MC GRPS = don't care
Follower
2.17 D2D MAIN CW
2.19 D2D REF1
(57.08 FOLLOWER CW SRC)
(57.06 REF 1 SRC)
57.01 LINK MODE = (1) Follower
57.03 NODE ADDRESS = 5 *
57.11 REF 1 MSG TYPE = (0) Broadcast *
57.12 REF1 MC GROUP = 5 *
57.13 NEXT REF1 MC GRP = don't care
57.14 NR REF1 MC GRPS = don't care
Follower
2.17 D2D MAIN CW
2.19 D2D REF1
57.01 LINK MODE = (1) Follower
57.03 NODE ADDRESS = 1
57.11 REF 1 MSG TYPE = don't care
57.12 REF1 MC GROUP = 2
57.13 NEXT REF1 MC GRP = don't care
57.14 NR REF1 MC GRPS = don't care
Follower
2.17 D2D MAIN CW
2.19 D2D REF1
57.01 LINK MODE = (1) Follower
57.03 NODE ADDRESS = 3
57.11 REF 1 MSG TYPE = don't care
57.12 REF1 MC GROUP = 4
57.13 NEXT REF1 MC GRP = don't care
57.14 NR REF1 MC GRPS = don't care

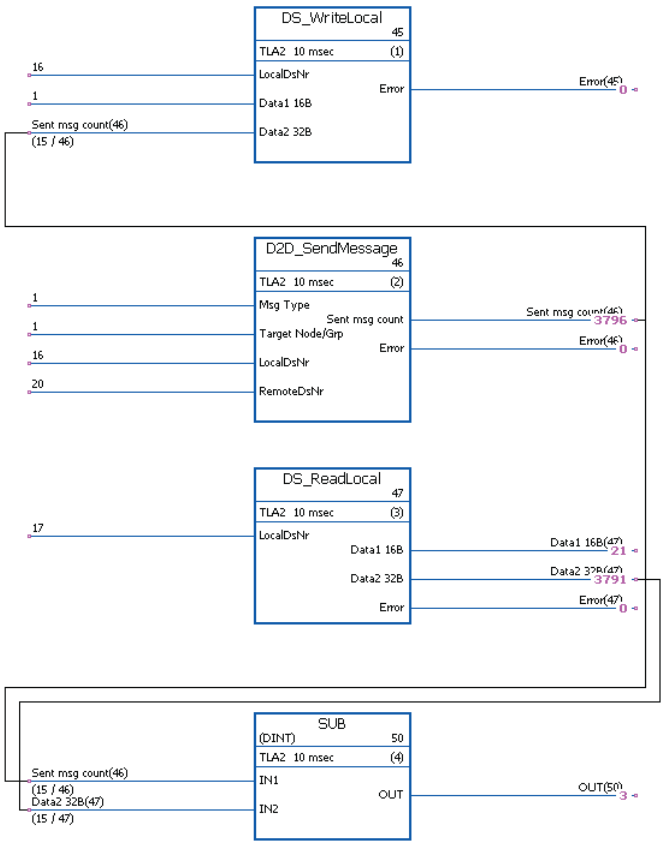
* Acknowledgement from last follower to master can be prevented by setting parameter 57.11 REF 1 MSG TYPE to (0) Broadcast (required because parameters 57.03 NODE ADDRESS and 57.12 REF1 MC GROUP are set to the same value). Alternatively, node/group addresses (parameters 57.03 NODE ADDRESS and 57.12 REF1 MC GROUP) could be set to non-equal values.

Examples of using standard function blocks in drive-to-drive communication

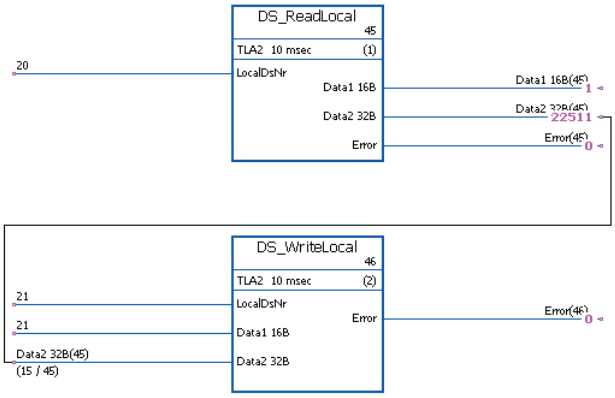
See also the descriptions of the drive-to-drive function blocks starting on page 246.

Example of master point-to-point messaging

Master



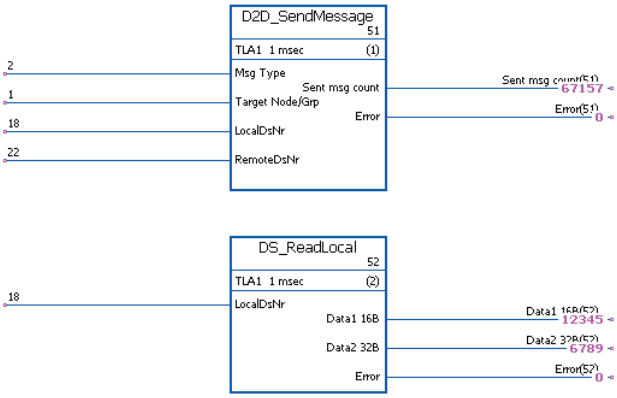
Follower (node 1)



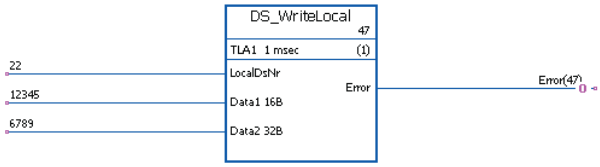
1. The master sends a constant (1) and the value of the message counter into follower dataset 20. Data is prepared to and sent from dataset 16.
2. The follower sends the received counter value and a constant (21) as a reply to the master.
3. The master calculates the difference of the latest message number and received data.

Example of read remote messaging

Master



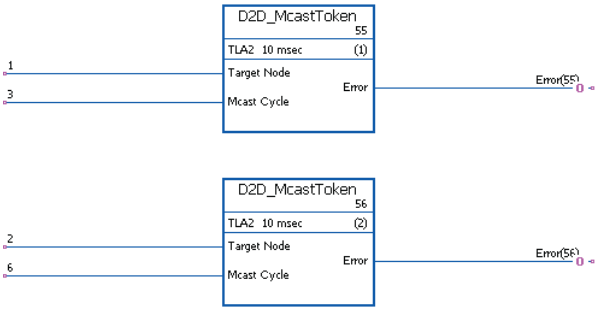
Follower (node 1)



1. The master reads the contents of the follower dataset 22 into its own dataset 18. Data is accessed using the DS_ReadLocal block.
2. In the follower, constant data is prepared into dataset 22.

Releasing tokens for follower-to-follower communication

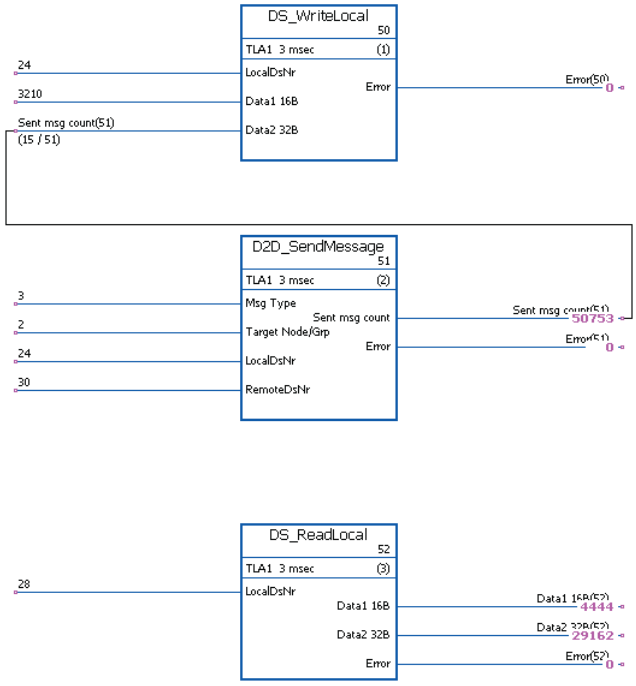
Master



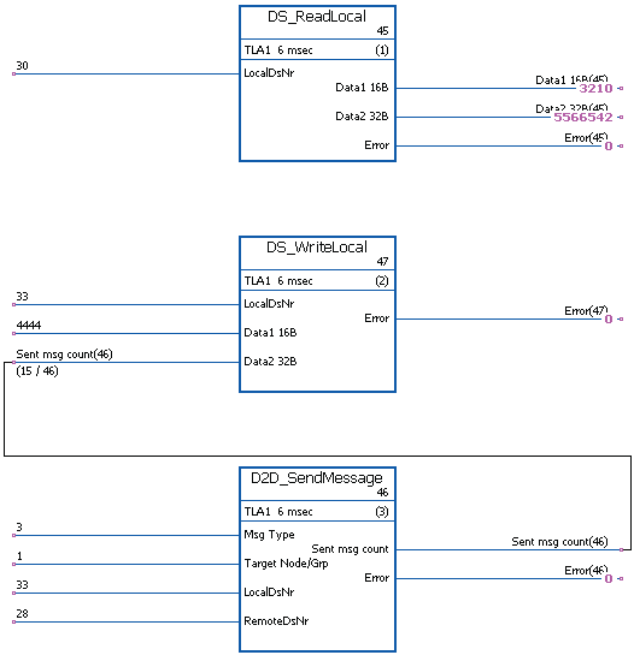
1. This drive-to-drive link consists of three drives (master and two followers).
2. The master operates as a “chairman”. Follower 1 (node 1) is allowed to send one message every 3 milliseconds. Follower 2 (node 2) is allowed to send one message every 6 milliseconds.

Example of follower point-to-point messaging

Follower 1 (node 1)



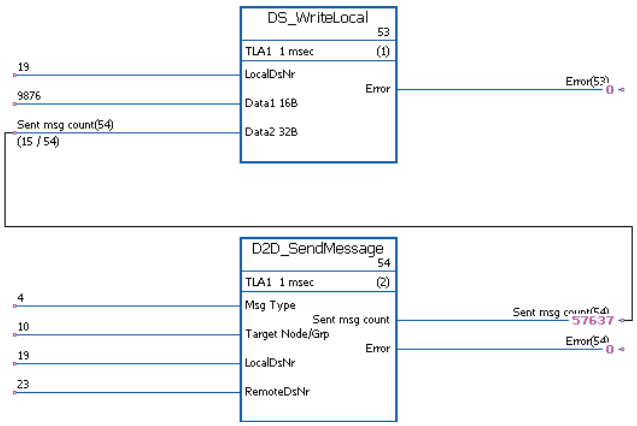
Follower 2 (node 2)



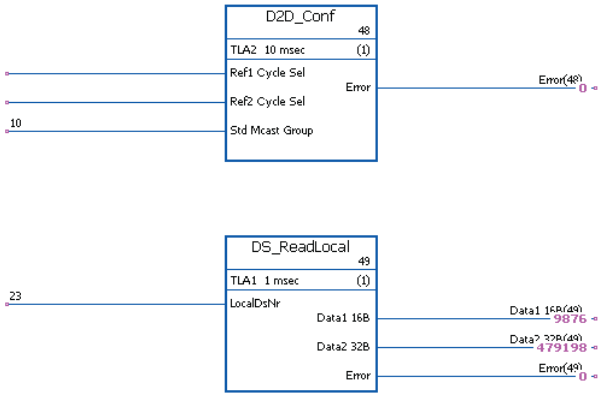
1. Follower 1 writes local dataset 24 to follower 2 dataset 30 (3 ms interval).
2. Follower 2 writes local dataset 33 to follower 1 dataset 28 (6 ms interval).
3. In addition, both followers read received data from local datasets.

Example of standard master-to-follower(s) multicast messaging

Master



Follower(s) in Std Mcast Group 10

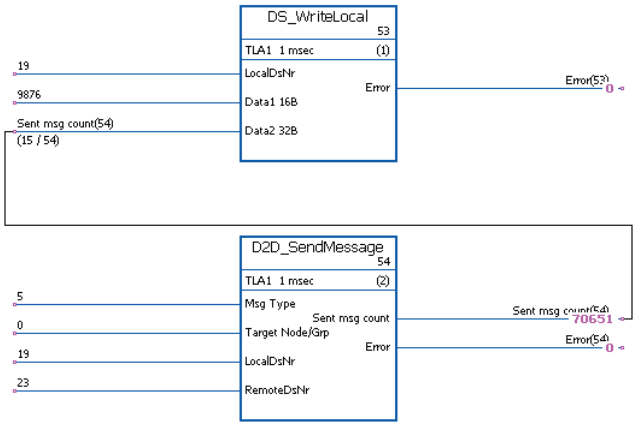


1. The master sends a constant (9876) and the value of the message counter to all followers in standard multicast group 10. The data is prepared into and sent from master dataset 19 to follower dataset 23.
2. Received data is read from dataset 23 of the receiving followers.

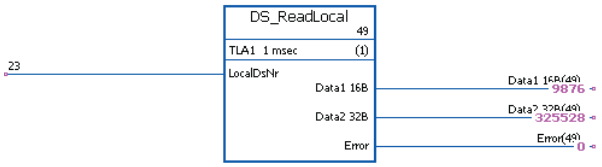
Note: The example application shown for Master above also applies to the sending follower in standard follower-to-follower multicasting.

Example of broadcast messaging

Master



Follower(s)



1. The master sends a constant (9876) and the value of the message counter to all followers. The data is prepared into and sent from master dataset 19 to follower dataset 23.
2. Received data is read from dataset 23 of the followers.

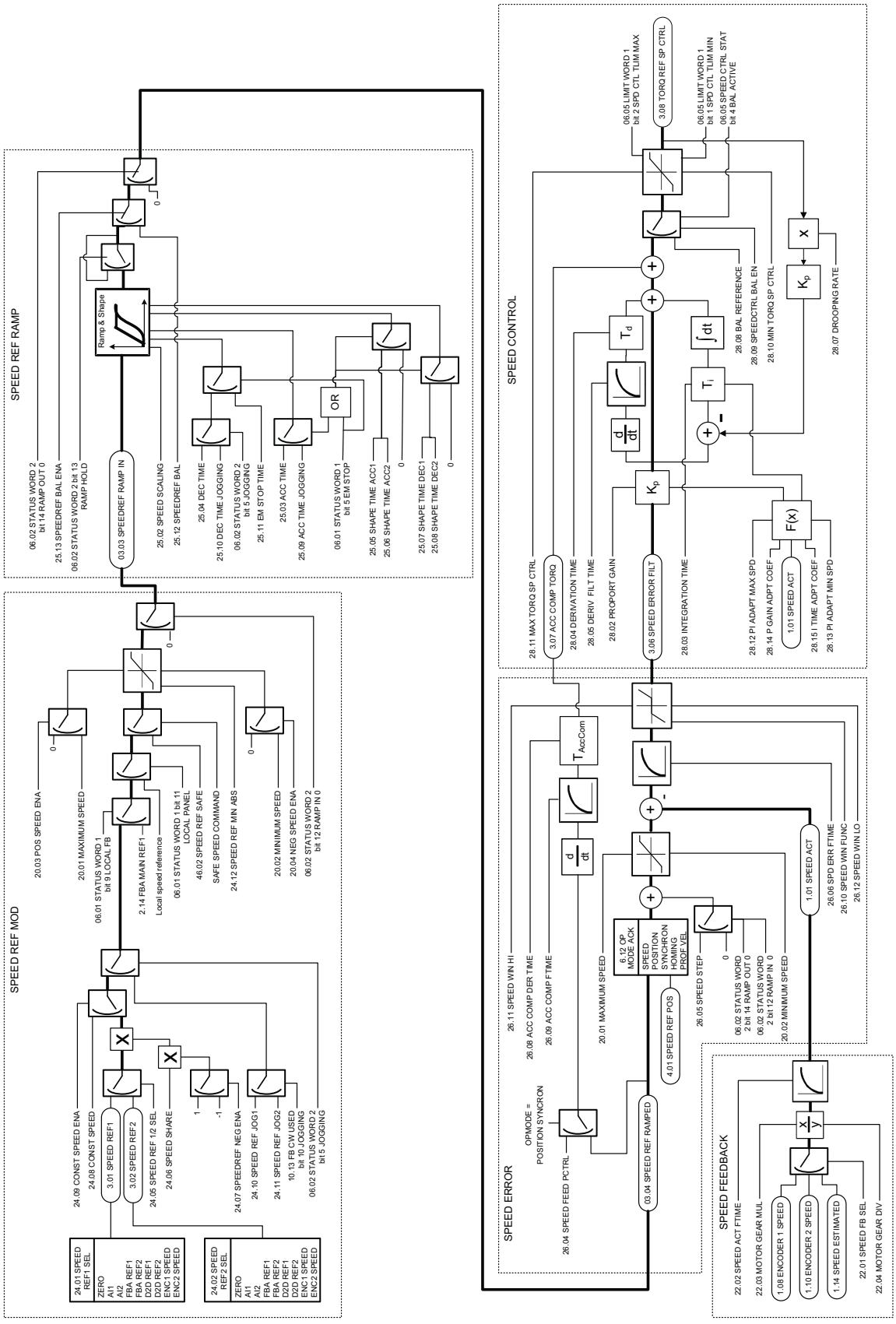
Note: The example application shown for Master above also applies to the sending follower in follower-to-follower broadcasting.

Appendix C – Control chain and drive logic diagrams

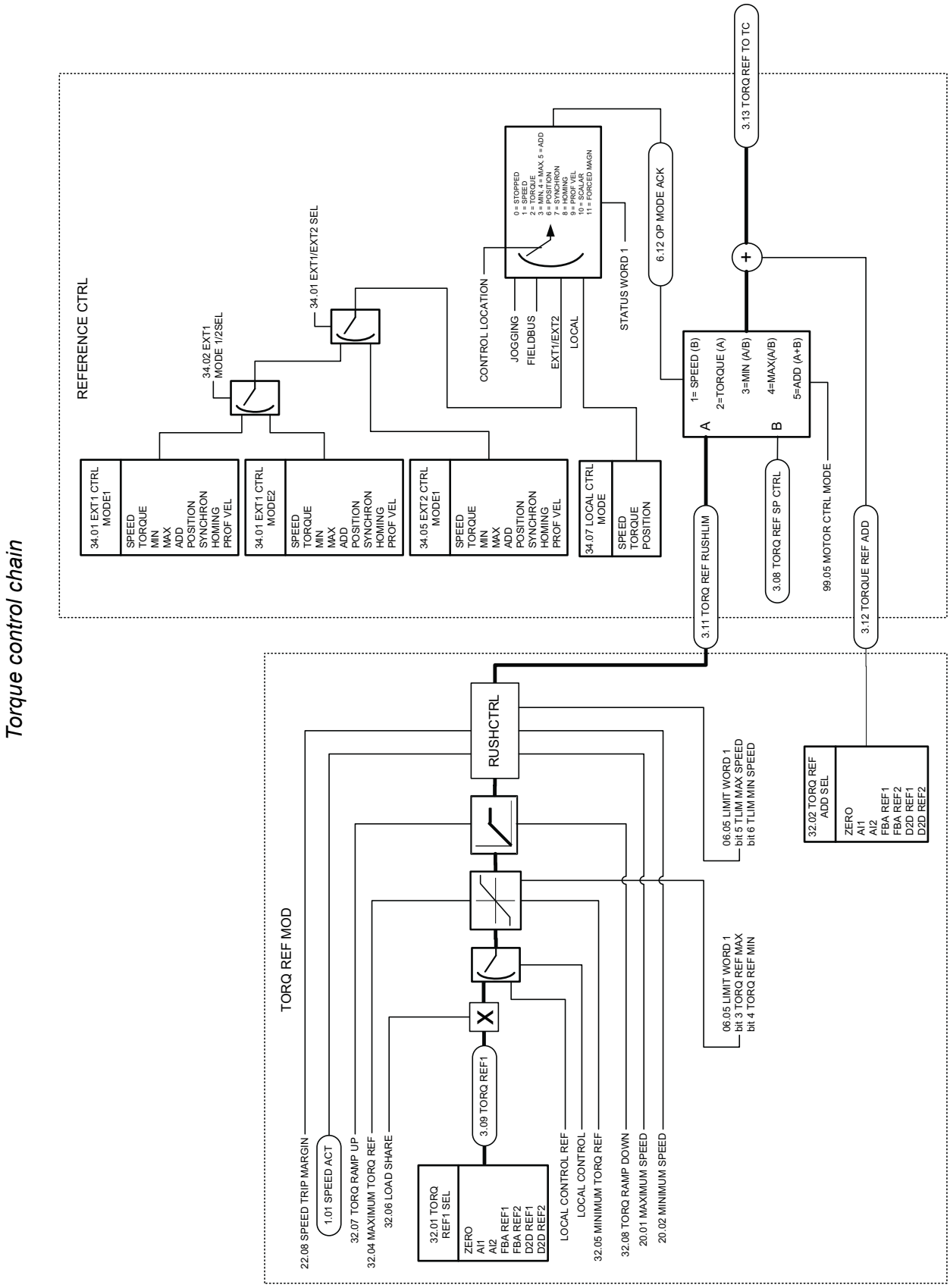
What this chapter contains

This chapter presents the drive control chain and logic.

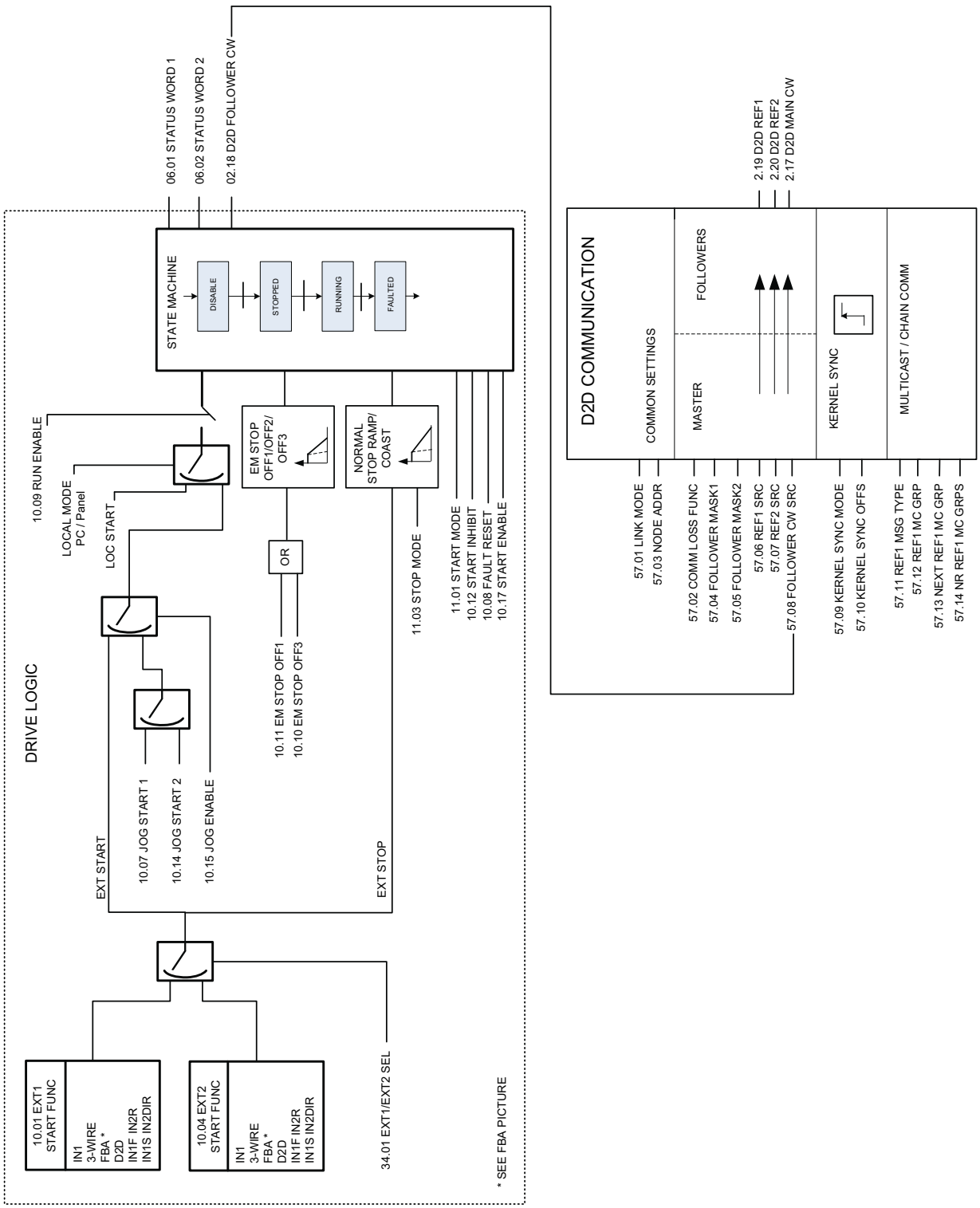
Speed control chain


SPEED REF MOD
SPEED REF RAMP
SPEED CONTROL
SPEED ERROR
SPEED FEEDBACK
Ramp & Shape
OR
F(x)
24.01 SPEED REF1 SEL
24.02 SPEED REF2 SEL
ZERO
AI1
AI2
FBA REF1
FBA REF2
D2D REF1
D2D REF2
ENC1 SPEED
ENC2 SPEED
20.03 POS SPEED ENA
20.01 MAXIMUM SPEED
06.01 STATUS WORD 1 bit 9 LOCAL FB
2.14 FBA MAIN REF1
Local speed reference
06.01 STATUS WORD 1 bit 11 LOCAL PANEL
46.02 SPEED REF SAFE
SAFE SPEED COMMAND
24.12 SPEED REF MIN ABS
20.02 MINIMUM SPEED
20.04 NEG SPEED ENA
06.02 STATUS WORD 2 bit 12 RAMP IN 0
24.09 CONST SPEED ENA
24.08 CONST SPEED
3.01 SPEED REF1
3.02 SPEED REF2
24.05 SPEED REF 1/2 SEL
24.06 SPEED SHARE
24.07 SPEEDREF NEG ENA
24.10 SPEED REF JOG1
24.11 SPEED REF JOG2
10.13 FB CW USED
bit 10 JOGGING
06.02 STATUS WORD 2 bit 5 JOGGING
06.02 STATUS WORD 2 bit 14 RAMP OUT 0
25.13 SPEEDREF BAL ENA
06.02 STATUS WORD 2 bit 13 RAMP HOLD
03.03 SPEEDREF RAMP IN
25.02 SPEED SCALING
25.12 SPEEDREF BAL
25.04 DEC TIME
25.10 DEC TIME JOGGING
06.02 STATUS WORD 2 bit 6 JOGGING
25.11 EM STOP TIME
25.03 ACC TIME
25.09 ACC TIME JOGGING
06.01 STATUS WORD 1 bit 5 EM STOP
25.05 SHAPE TIME ACC1
25.06 SHAPE TIME ACC2
25.07 SHAPE TIME DEC1
25.08 SHAPE TIME DEC2
06.05 LIMIT WORD 1 bit 2 SPD CTL TLIM MAX
3.08 TORQ REF SP CTRL
06.05 LIMIT WORD 1 bit 1 SPD CTL TLIM MIN
06.05 SPEED CTRL STAT bit 4 BAL ACTIVE
28.11 MAX TORQ SP CTRL
3.07 ACC COMP TORQ
28.04 DERIVATION TIME
28.05 DERIV FILT TIME
28.02 PROPORT GAIN
3.06 SPEED ERROR FILT
28.03 INTEGRATION TIME
28.08 BAL REFERENCE
28.09 SPEEDCTRL BAL EN
28.10 MIN TORQ SP CTRL
28.12 PI ADAPT MAX SPD
28.14 P GAIN ADPT COEF
1.01 SPEED ACT
28.15 I TIME ADPT COEF
28.13 PI ADAPT MIN SPD
28.07 DROOPING RATE
26.11 SPEED WIN HI
26.08 ACC COMP DER TIME
26.09 ACC COMP F TIME
20.01 MAXIMUM SPEED
6.12 OP MODE ACK
SPEED
POSITION
SYNCHRON
HOMING
PROF VEL
4.01 SPEED REF POS
26.05 SPEED STEP
06.02 STATUS WORD 2 bit 14 RAMP OUT 0
06.02 STATUS WORD 2 bit 12 RAMP IN 0
20.02 MINIMUM SPEED
26.04 SPEED FEED PCTRL
OPMODE = POSITION SYNCRON
03.04 SPEED REF RAMPED
26.06 SPD ERR FTIME
26.10 SPEED WIN FUNC
26.12 SPEED WIN LO
22.02 SPEED ACT FTIME
22.03 MOTOR GEAR MUL
1.08 ENCODER 1 SPEED
1.10 ENCODER 2 SPEED
1.14 SPEED ESTIMATED
22.01 SPEED FB SEL
22.04 MOTOR GEAR DIV

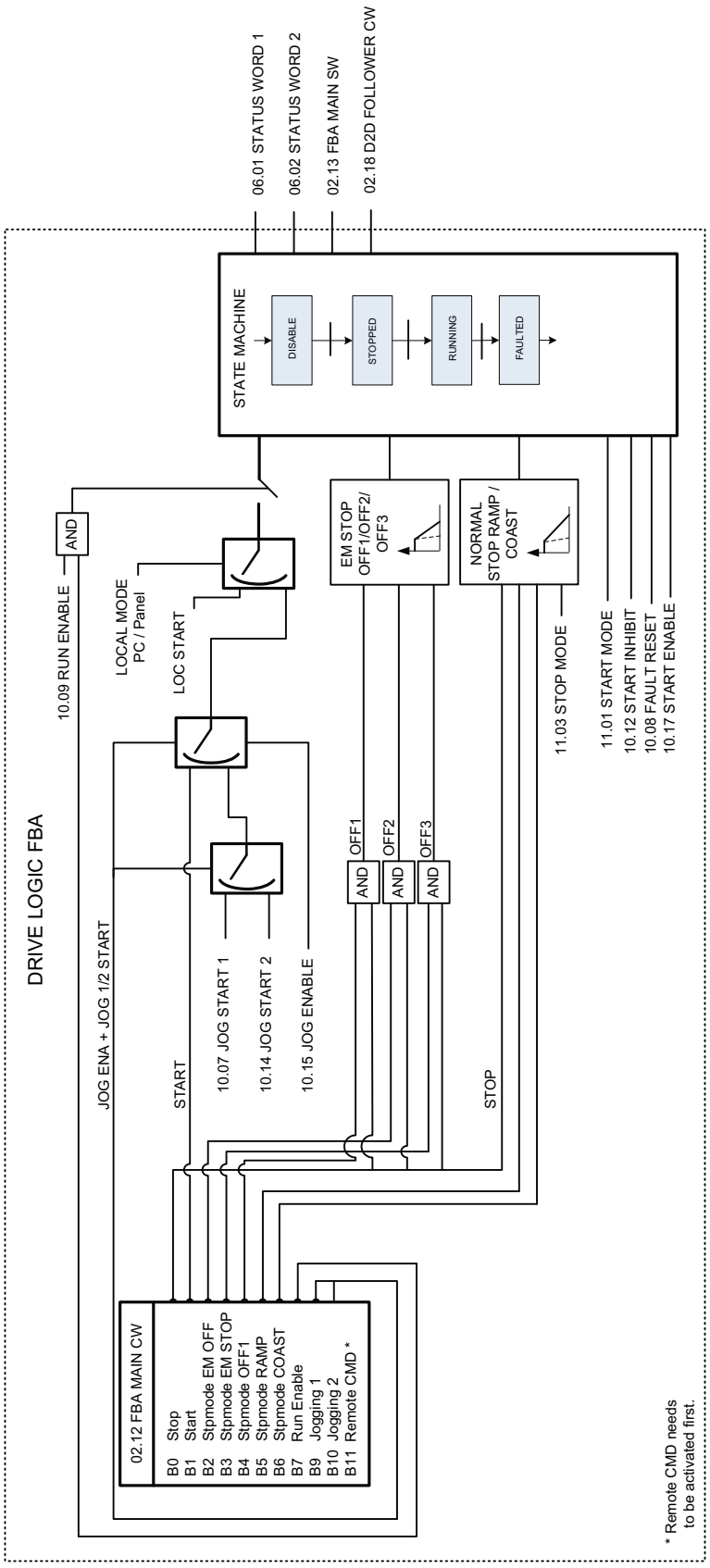
Torque control chain



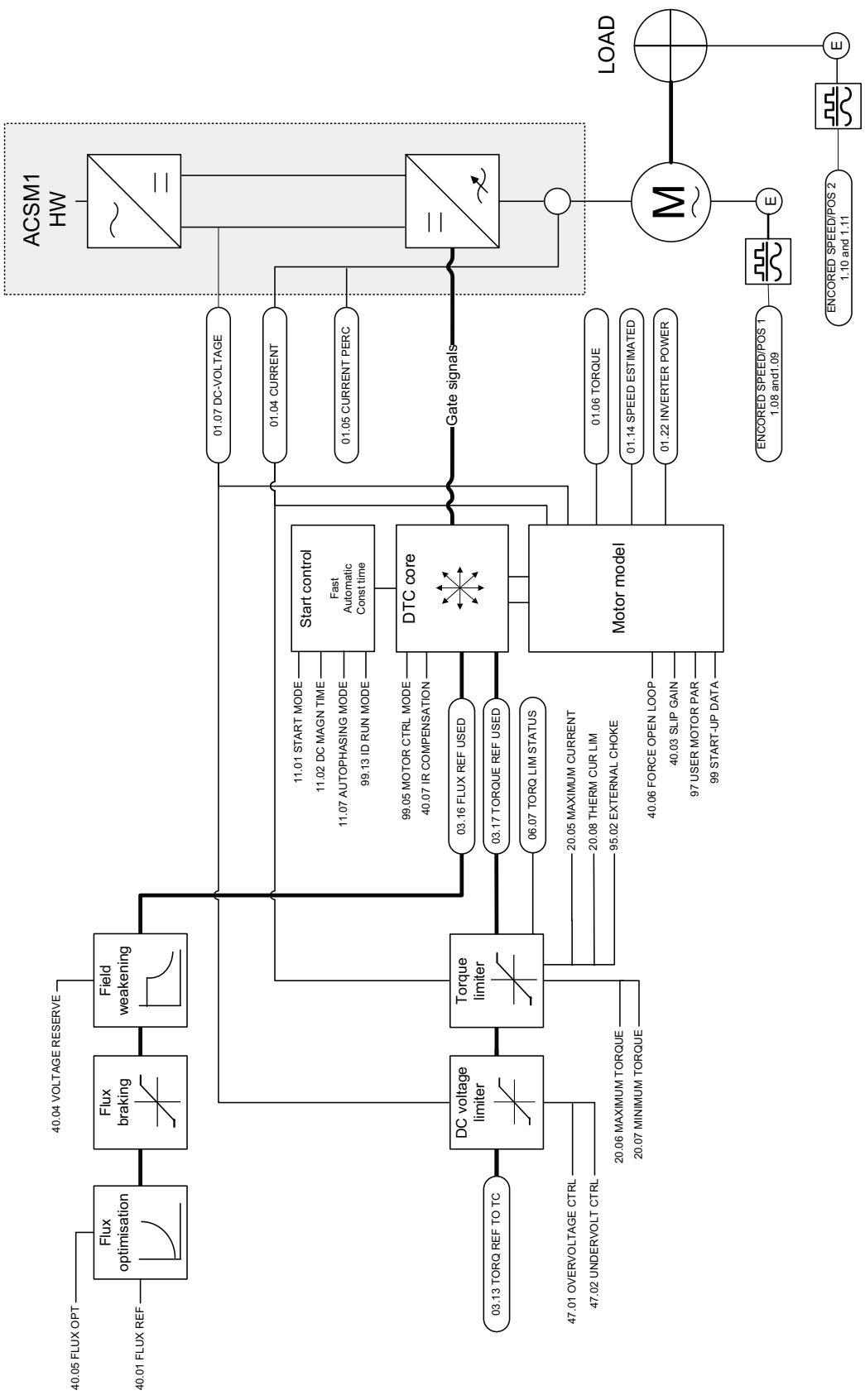
Drive logic 1



Drive logic 2 (Fieldbus interface)



DTC motor control



3AFE68848261 REV F / EN
EFFECTIVE: 2009-12-21

ABB Oy
AC Drives
P.O. Box 184
FI-00381 HELSINKI
FINLAND
Telephone +358 10 22 11
Fax +358 10 22 22681
Internet http://www.abb.com

ABB Inc.
Automation Technologies
Drives & Motors
16250 West Glendale Drive
New Berlin, WI 53151
USA
Telephone 262 785-3200
800-HELP-365
Fax 262 780-5135

ABB Beijing Drive Systems Co. Ltd.
No. 1, Block D, A-10 Jiuxianqiao Beilu
Chaoyang District
Beijing, P.R. China, 100015
Telephone +86 10 5821 7788
Fax +86 10 5821 7618
Internet http://www.abb.com